Baze de date

Universitatea “Transilvania” din Brasov

Lect.dr. Costel Aldea

costel.aldea@gmail.com

Observatii (teme / prezente)

I)	min 5 prezente curs	min 9 prezente laborator	
	1 absenta	=	a) referat 2-3 pagini (minim 2 surse bibliografice)
		(la alegere a,b sau c)	b) o prezentare de 10-15 slideuri (minim 2 surse bibliografice)
		(din curs 11)	c) implementare operatii CRUD / tabel - folosind prj test https://github.com/aclblaj/vdb
II)	Proiect BD	mail la	costel.aldea@gmail.com + laborant grupa
III)	Examen	Consultatii	
	zz	-2 (cu doua zile inainte de examen)	zz-2
	26.ian	-2 (cu doua zile inainte de examen)	24.ian
IV)	Subiecte		

ACID vs. BASE

La **sistemele de baze de date relaționale se folosește un sistem tranzacțional de gestiune: operațiile de modificare a bazei de date sunt grupate în tranzacții. La aceste sisteme o** tranzacție respecta **modelul ACID:**

- **Atomica:** tranzacția este tratata ca o singura unitate (daca o singura operație nu se poate executa, atunci se anulează efectul întregii tranzacții)
- **Consistenta:** baza de date este consistenta la sfârșitul execuției tranzacției (respecta regulile de integritate memorate)
- **Izolata:** pe timpul execuției unei tranzacții T alte tranzacții nu pot modifica datele gestionate de tranzacția T
- **Durabila:** daca o tranzacție s-a terminat de executat, atunci sistemul asigura ca ea nu mai trebuie re-executata în cazul unor erori.

Exist foarte multe aplicații unde un astfel de sistem tranzacțional este foarte important.

La **sistemele NoSQL modelul ACID este greu sa fie respectat (mai ales din cauza distribuirii si** replicării), și atunci el se înlocuiește cu **modelul BASE:**

- **Basic Availability:** toti clienții primesc un răspuns la o interogare (în loc de a folosi o singura sursa de date, colecția de date este replicata si distribuita, deci undeva în rețea este posibil sa existe datele căutate)
- **Soft State:** consistența bazei de date nu este verificata de SGBD, ea trebuie sa fie asigurata de clientul (programul) care are dreptul de modificare a bazei de date
- **Eventual Consistency:** baza de date poate sa se afle într-o stare de *inconsistența* (exista valori diferite ale aceleași date), dar *se presupune* ca în viitor datele vor ajunge într-o stare de consistență. Propagarea modificărilor la replicile datei va fi efectuată în viitor.

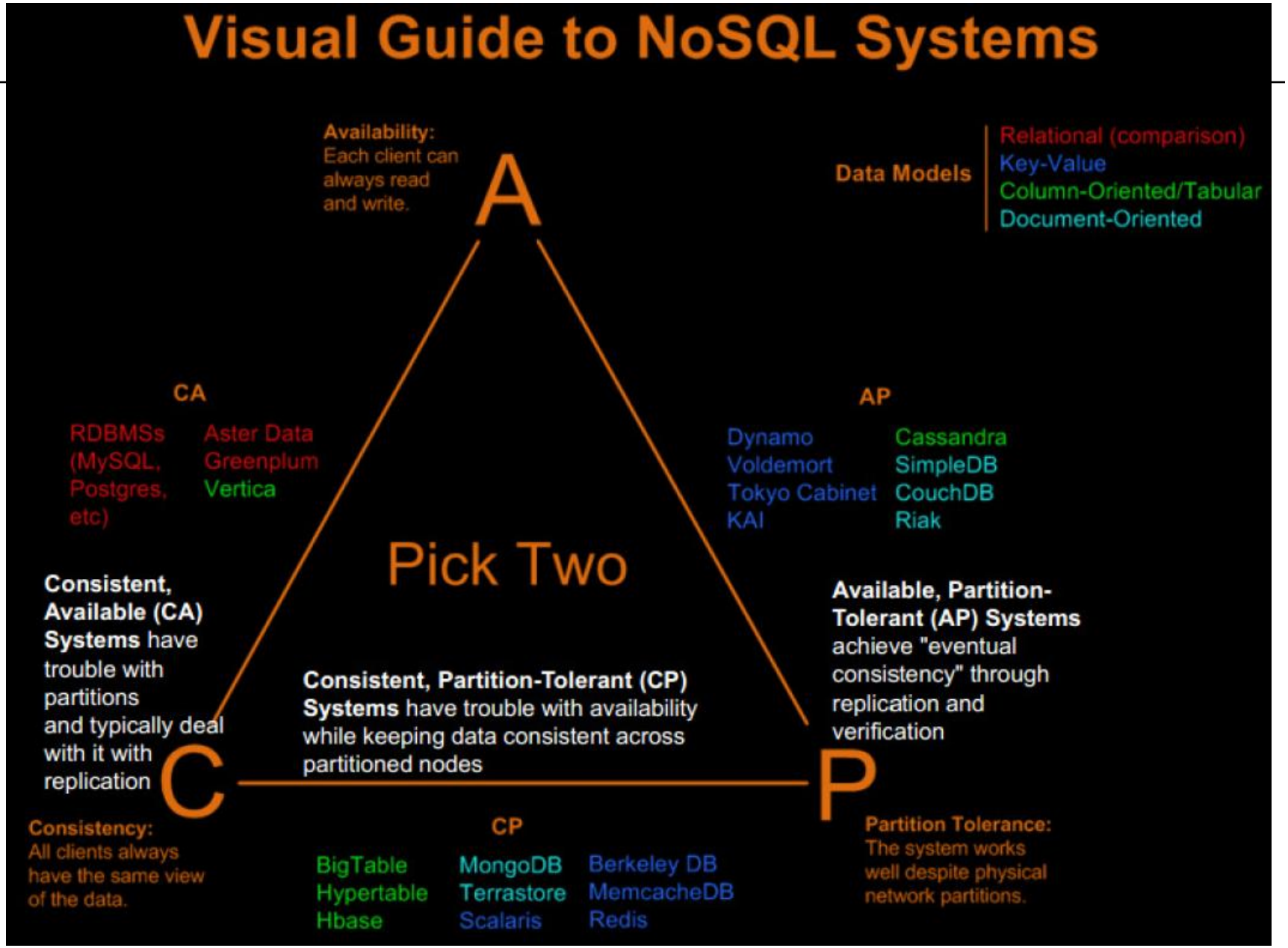
Teorema CAP

- O conjectura (a lui Brewer) a fost demonstrată în [GiLy02] si denumita **Teorema CAP:**

Este imposibil pentru un serviciu web sa ofere simultan următoarele trei facilitai:

- Consistența (**Consistency)**
- Disponibilitate la cereri (**Availability)**
- Facilitați pentru partiție (**Partition)**

- Folosind aceasta teorema, sistemele de baze de date se pot împarți în trei categorii (clase) după proprietățile pe care le au: **CA, AP, CP.** Deoarece sistemele NoSQL trebuie sa permită partiționarea, acestea sunt incluse în una din clasele:
  - CP (Consistency + Partition): au un grad mai ridicat de consistența, în defavoarea disponibilitătii
  - AP (Availability + Partition): au un grad mai ridicat de disponibilitate, în timp ce restricțiile cerute pentru consistentă s-au restrâns (sau chiar eliminat)
- In [Hurst] se da următoarea imagine cu privire la categoriile (clasele) amintite mai sus.


Visual Guide to NoSQL Systems
Availability:
Each client can always read and write.
A
Data Models
Relational (comparison)
Key-Value
Column-Oriented/Tabular
Document-Oriented
CA
RDBMSs (MySQL, Postgres, etc)
Aster Data
Greenplum
Vertica
AP
Dynamo
Voldemort
Tokyo Cabinet
KAI
Cassandra
SimpleDB
CouchDB
Riak
Pick Two
Consistent, Available (CA) Systems have trouble with partitions and typically deal with it with replication
Consistent, Partition-Tolerant (CP) Systems have trouble with availability while keeping data consistent across partitioned nodes
Available, Partition-Tolerant (AP) Systems achieve "eventual consistency" through replication and verification
C
P
Consistency:
All clients always have the same view of the data.
CP
BigTable
Hypertable
Hbase
MongoDB
Terrastore
Scalaris
Berkeley DB
MemcacheDB
Redis
Partition Tolerance:
The system works well despite physical network partitions.

CEP (Complex Event Processing)

- Prelucrare complexă de evenimente (CEP) consta in utilizarea tehnologiei pentru a prezice evenimente care ar putea rezulta din seturi specifice de factori de baza.
- CEP identifică și analizează relațiile cauză-efect între evenimente în timp real, care să permită să se efectueze în mod proactiv o acțiune în mod eficient ca răspuns la scenarii specifice.
- CEP este o paradigmă în evoluție - conceput inițial în anii 1990 de către Dr. David Luckham la Universitatea Stanford.

Descriere conceptuală

- Printre mii de evenimente de intrare, un sistem de monitorizare poate, de exemplu, să primească următoarele trei din aceeași sursă:
  - clopotele bisericilor suna.
  - apariția unui om într-un smoching cu o femeie într-o rochie albă care curge.
  - orez care zboară prin aer.
- Din aceste evenimente sistemul de monitorizare poate deduce un eveniment complex: o nunta. CEP este o tehnică ce ajută la descoperirea evenimentelor complexe prin analiza si corelarea altor evenimente: clopotele, bărbatul și femeia în costum de nunta si orezul zboară prin aer.
- CEP se bazează pe o serie de tehnici, inclusiv:
  - Detectare eveniment model
  - Abstractizarea unui eveniment
  - Filtrarea evenimentelor
  - Agregare eveniment și transformarea
  - Modelarea unor ierarhii de evenimente
  - Relatii de detectare (cum ar fi cauzalitatea, membru sau sincronizare [dezambiguizare necesara]) între evenimente
  - Abstractizare procese determinate de un eveniment
- Aplicații CEP comerciale include:
  - tranzacționare stocuri algoritmic,
  - detectarea fraudelor credit-card,
  - monitorizare activității și monitorizare de securitate.

Utilitatea CEP

- CEP este utilizat în
  - securitatea politicii riscului în management,
  - managementul relațiilor cu clienții (CRM),
  - servere de aplicații și middleware.
- Un aspect important al CEP este monitorizarea activității de afaceri (BAM), utilizarea tehnologiei pentru a defini în mod proactiv și să analizeze oportunitățile critice și riscurile dintr-o întreprindere.
- CEP este deosebit de eficientă în situații care implică numeroși factori care interacționează în moduri variabile, cum ar fi de investiții și medii de creditare pentru instituțiile financiare.
- CEP poate fi de asemenea utilizat în managementul amenințărilor in rețelele de comunicații.

Software și subsisteme

- O mare varietate de programe software comerciale de prelucrare eveniment sunt disponibile pentru arhitecți și dezvoltatori care construiesc aplicații de procesare de evenimente. Acestea sunt uneori denumite platforme de procesare de evenimente, sisteme complexe de procesare de evenimente (CEP), sistemele de prelucrare fluxuri de evenimente (ESP), sau platforme fluxuri de calcul distribuit (DSCPs).
- Câteva exemple sunt:
  - Apache Samza
  - Apache Spark
  - Codehaus / lui EsperTech Esper, Nesper
  - DataTorrent RTS (real-time Streaming)
  - FujitsuInterstage Big Data eveniment complex de procesare Server
  - IBM InfoSphere Streams
  - IBM Decizia Operațional Management (ODM)
  - Microsoft StreamInsight
  - Oracle Eveniment Procesor
  - RedHat Drools Fusion / JBoss Enterprise BRMS
  - SAP Eveniment Stream Processor
  - SAS DataFlux
  - ScaleOut Software
  - SQLStream s-Server
  - etc.

Continuitate

- Utilizarea CEP este în creștere rapidă, deoarece CEP, într-un sens tehnic, este singura modalitate de a extrage informații de la fluxurile de evenimente in timp real sau timp aproape real. Sistemul trebuie să proceseze datele legate de eveniment mai mult sau mai puțin pe măsură ce acesta intra, astfel încât (contra)acțiunea corespunzătoare pot fi luata rapid.

NLP

- Procesarea limbajului natural (NLP-Natural Language Processing) este un domeniu derivat din
  - informatică,
  - inteligență artificială, și
  - lingvistică computațională specializată pe interacțiunile dintre calculatoare și limbaje (naturale) umane.
- NLP este legat de zona de interacțiune om-calculator. Multe provocări în NLP implica înțelegerea limbajului natural, care să permită computerelor să obțină un înțeles de la limbajului uman sau natural, iar altele presupun generarea limbajului natural.

Inceputurile NLP

- Istoria NLP începe din anii 1950, cu toate că pot fi găsite încercări și mai devreme de atât.
- În 1950, Alan Turing a publicat un articol intitulat „Tehnica și inteligența calculatoarelor„(“Computing Machinery and Intelligence”) care a propus ceea ce se numește acum testul Turing ca un criteriu de inteligență.

Testul Turing

- Testul Turing este un test al capacității unei mașini de a expune inteligent comportamentul echivalent, sau imposibil de distins de, cea a unui om.
- Alan Turing a propus ca un evaluator uman să judece conversația lingvistică naturală dintre un om și o mașină proiectată să genereze răspunsuri umane.
- Evaluatorul ar fi conștient de faptul că unul dintre cei doi parteneri din conversație este o mașină, și toți participanții vor fi separați unul de celălalt.
- Conversația ar fi limitata la un canal scris, cum ar fi o tastatură de calculator și ecran, astfel încât rezultatul să nu trebuiască să depindă de capacitatea aparatului de a reda cuvintele ca discurs.
- În cazul în care evaluatorul nu poate distinge aparatul de la om (Turing inițial a sugerat că mașina ar convinge un om de 70% din timpul de după cinci minute de conversație), aparatul se spune că a trecut testul.
- Testul nu verifică capacitatea de a da răspunsuri corecte la întrebări, ci doar capacitatea mașinii de a da răspunsuri cât de cât umane.

NLP și ELIZA

- Un succes notabil al sistemelor NLP dezvoltate în anii 1960 au fost SHRDLU, un sistem de limbaj natural care lucrează în „blocks worlds", cu vocabulare restrânse
- ELIZA, o simulare a unui psihoterapeut Rogerian, scris de Joseph Weizenbaum între 1964 -1966. Aceasta chiar daca nu utiliza aproape nici o informație despre gândirea umană sau emoție, ELIZA răspundea uneori ca o ființă umană.
- Atunci când "pacientul" a depășit baza de cunoștințe foarte mica a acesteia, ELIZA putea oferi un răspuns generic, de exemplu, ca răspuns la „Mă doare capul" ea ar fi spus "De ce te doare capul?".

Introducerea algortimilor în NLP

- Până în anii 1980, cele mai multe sisteme NLP-au avut la bază seturi complexe de reguli scrise de mână. Pornind de la sfârșitul anilor 1980, cu toate acestea, a existat o revoluție în NLP cu introducerea unor algoritmi pentru procesarea limbajului.
- Cercetarile recente s-au concentrat tot mai mult pe algoritmi de învățare nesupravegheate și semi-supravegheate. Astfel de algoritmi sunt capabili să învețe de la date care nu au fost adnotate de mână cu răspunsurile dorite, sau folosind o combinație de date adnotate și non-adnotate. În general, această sarcină este mult mai dificilă decât de învățare supervizată, și de obicei produce rezultate mai puțin precise pentru o anumită cantitate de date de intrare. Cu toate acestea, există o cantitate enormă de date care nu sunt adnotat disponibile (inclusiv, printre altele, întregul conținut al World Wide Web), care poate oferi de multe ori pentru rezultatele inferioare.

NLP prin “machine learning”

- Algoritmi moderni NLP se bazeaza pe “machine learning”, mai ales mașinile de statistică.
- Paradigma de programare a mașinilor este diferită de cea a majorității încercărilor anterioare de procesare a limbajului. Implementări anterioare de sarcini de procesare a limbajului au implicat codificarea prin scrierea de mână a unor seturi mari de reguli.
- Paradigma “machine learning” solicită în schimb, folosirea unor algoritmi de învățare generali, să învețe mașina în mod automat astfel de norme, prin analiza de corpora mari, tipice din lumea reală.
- Un corpus (plural, "corpora") este un set de documente (sau uneori, fraze individuale) care au fost adnotate de mână cu valorile corecte pentru a fi învățate.

Task-uri în NLP

- Mai jos este o listă a sarcinilor cel mai frecvent cercetate în NLP. Unele dintre aceste task-uri au aplicație directă din lumea reala, în timp ce altele mai frecvent servesc ca subactivități care sunt utilizate pentru a ajuta la rezolvarea task-urilor mai mari.
- Ceea ce distinge aceste task-uri de la alte task-uri potențiale și reale NLP nu este numai volumul de cercetare dedicat acestora, dar și faptul că pentru fiecare dintre este de obicei o setare bine definită de probleme, o măsurătoare standard pentru evaluarea sarcinii, corpora standard la care task-ul poate fi evaluat și concursuri dedicate task-ului specific.

NLP - Sintetizarea automata

- Sintetizarea automata – crearea unei versiuni mai scurte a unui text de către un program. Rezultatul operației conține totuși majoritatea punctelor importante din textul original. Se folosește in special pentru articole financiare din ziare, sau motoarele de căutare (Google).

NLP - Coreference

- Coreference – apare atunci când doua sau mai multe cuvinte dintr-un text se refera la același obiect/persoana (au aceeași referința).

ex. Bill said he would come.

NLP - Analiza discursului

- Analiza discursului – termen general pentru un numar de incercari de analiza:
  - a scrisului,
  - vorbitului
  - sau alta utilizare a limbii.
- Obiectele (discursul, scrisul, conversatia sau evenimentul de comunicare) sunt definite prin secvente de propozitii, fraza.

NLP - Traducerea automata

- Traducerea automata – traduce text dintr-o limba vorbita intr-o alta limba.
- Este cea mai complicata parte deoarece necesita cunostinte despre gramatica a celor doua limbi, de semantica, despre viata reala.

NLP - Segmentarea la nivel morfologic

- Segmentarea la nivel morfologic – desparte cuvintele in morfeme (cea mai mica parte a unui cuvant care poarta o informatie) individuale si identifica clasa morfemului.
- Dificultatea se bazeaza pe complexitatea structurii cuvântului din limba determinata.

NLP - Identificarea entitatii

- Identificarea entitatii (Named-entity recognition)
- recunoaste cuvintele dintr-un dialog care se refera la persoane sau locuri si la ce se refera exact acel cuvant (persoana, loc, organizatie).

NLP - Generarea limbajului natural

- Generarea limbajului natural – converteste informatia din baza de date a calculatorului in limbaj uman.

NLP - Intelegerea limbajului natural

- Intelegerea limbajului natural – transforma parti din text intr-un limbaj care este mai usor de procesat de calculator – precum logica first-order

NLP -

- Recunoașterea optica a caracterelor – pe baza unei imagini care reprezinta un text printat.
- Determinarea părților de vorbire – determina daca un cuvânt este substantiv, verb, adjectiv. Ex: read a book (book – noun) ; to book a flight (book – verb)
- Parsare – analiza gramatica a unei propoziții
- Răspunderea la întrebări – in limbajul uman, exista întrebări cu răspuns simplu (care este capitala Canadei?) sau cu răspuns deschis (care este sensul vieții?)

NLP -

- Aflarea relatiilor – de exemplu cine este sotul unei persoane (Ana este casatorita cu Dorel.)
- Segmentarea frazelor – de obicei sunt folosite semne de punctuatie intre propozitii
- Recunoasterea unui discurs oral – reprezentarea textuala a unui discurs oral; foloseste Segmentarea unui discurs oral – care desparte discursul in cuvinte
- Extragerea informatiei din text – extragerea informatiei semantice din text; foloseste determinarea numelor, coreference, aflarea relatiilor

Bibliografie

- [GiLy02] Seth Gilbert, Nancy Lynch, Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services, ACM SIGACT News, Volume 33 Issue 2, June 2002, Pages 51-59, (sau BrewersConjecture-SigAct.pdf)
- [Hurst] Nathan Hurst, Visual Guide to NoSQL Systems