

Napredni modeli i baze podataka

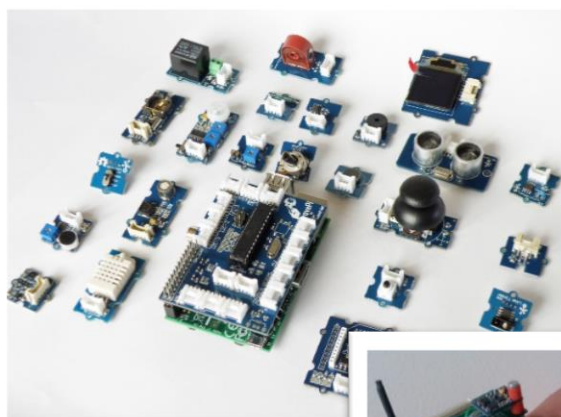
Predavanja
Siječanj 2016.

10. Tokovi podataka

- **Sustavi za upravljanje tokovima podataka (SUTP)**
 - Motivacija
 - Tokovi podataka – pojmovi i modeli
 - SUBP vs. SUTP
 - Arhitektura SUTP
 - Aplikacijski zahtjevi
 - Upiti i upitni jezici
- **Distribuirana obrada tokova podataka (Apache Spark Streaming)**
 - Povijest ukratko
 - Apache Spark
 - Osnovne karakteristike
 - Resilient Distributed Datasets i SparkContext
 - Distribuirana obrada
 - Komponente
 - Apache Spark Streaming

Sustavi za upravljanje tokovima podataka (SUTP)

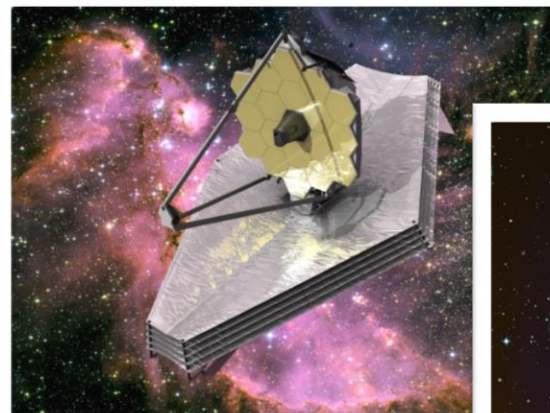
Motivacija



Senzorske mreže



Nosivi senzori



Satelitska mjerenja



Pametni telefoni



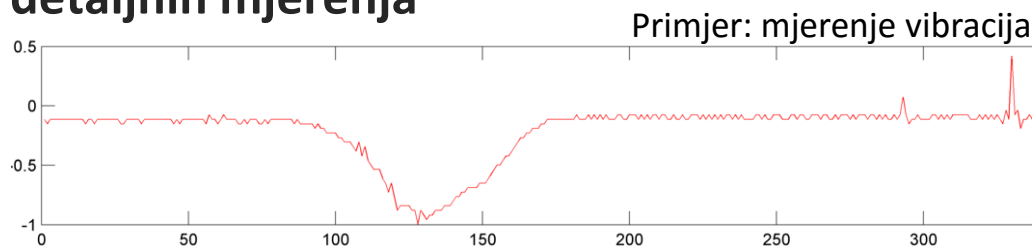
Telekomunikacije

Motivacija za SUTP

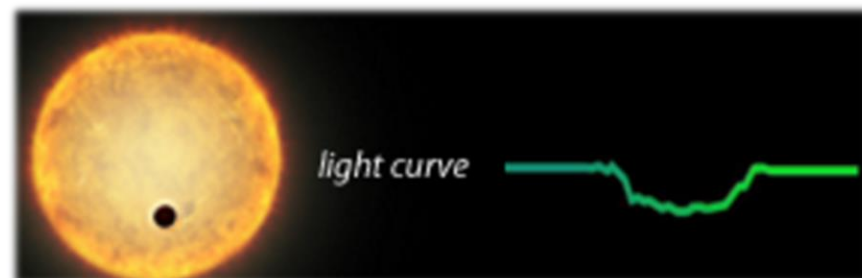
- Ogromne količine zanimljivih/korisnih podataka
 - Postavljanje točaka za promatranje **transakcijskih podataka**, npr.
 - AT&T pozivi: ~300M n-torki/dan
 - AT&T IP infrastruktura: ~10B IP tokova/dan

- Automatizirano generiranje jako **detaljnih mjerenja**

- Senzorske mreže: ogromni broj mjernih točaka
(eng. *measurement points*)



- [Kepler](#): detaljna satelitska mjerenja svjetlosnog intenziteta zvijezda – potraga za egzoplanetima
 - Uskoro (2018): [James Webb Space Telescope](#)
 - još više podataka!



- Sumarno
 - *Velike količine sirovih (eng. raw) podataka, pristižu velikom brzinom*
 - *Analiza potrebna čim je prije moguće!*

Tokovi podataka - pojmovi

- **Tok podataka** – (potencijalno neograničen) slijed n-torki (eng. *tuples*)
- **Transakcijski tokovi podataka:** logiranje interakcija između entiteta
 - Telekomunikacije: zapisi o telefonskim pozivima između korisnika
 - Web: zapisi o pristupima klijenata prema resursima na serverima
 - Kreditne kartice: zapisi o transakcijama između kupaca i trgovaca
 - ...
- **Tokovi podataka temeljeni na mjerenjima:** praćenje evolucije stanja entiteta
 - Senzorske mreže: mjerenje fizičkih fenomena (temperatura, vlažnost, seizmičke aktivnosti, ...), promet, ljudska aktivnost, ...
 - IP mreže: mrežni promet na sučeljima IP usmjerivača

Modeli tokova podataka

- Ulazni tok a_1, a_2, \dots pristiže slijedno (podatak za podatkom) te opisuje *signal* $\mathbf{A} : [1 \dots N] \rightarrow \mathbb{R}^2$
 - Modeli se razlikuju prema načinu na koji elementi a_i opisuju signal \mathbf{A}
 - Mogući višestruki tokovi na ulazu te višedimenzionalni signali

1. Time Series Model (vremenske serije)

- Svaki element a_i je jednak $\mathbf{A}[i]$
- Elementi se pojavljuju u rastućem slijedu varijable i
- Npr. IP mrežni promet svakih n minuta, vrijednost valute (npr. BitCoin) svakih n sekundi ...



Modeli tokova podataka (2)

2. Cash Register Model

- Elementi a_i su inkrementi elemenata $\mathbf{A}[j]$
- Tijekom vremena različiti elementi a_i mogu povećati vrijednost istog elementa $\mathbf{A}[j]$
- Npr. nadgledanje IP adresa koje pristupaju web poslužitelju (istom web poslužitelju $\mathbf{A}[j]$ pristupa se s različitim IP adresa a_i)



3. Turnstile model

- Elementi a_i su ažuriranja elemenata $\mathbf{A}[j]$
- Tijekom vremena različiti elementi a_i mogu povećati ili smanjiti vrijednost istog elementa $\mathbf{A}[j]$
- Za promatranje potpuno dinamičnih pojava (postoji *insert* i *delete*)
- Najopćenitiji model

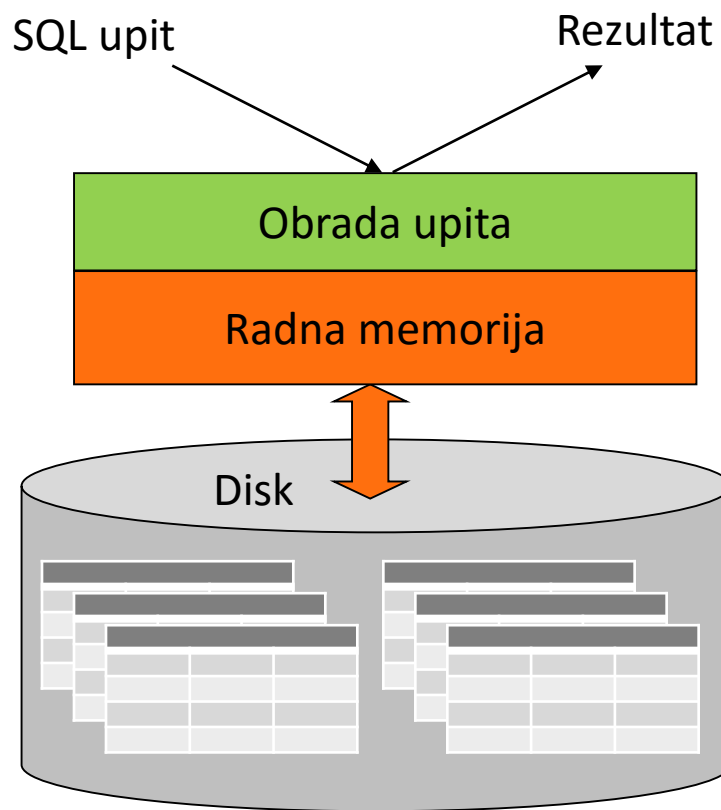


Upravljanje podacima

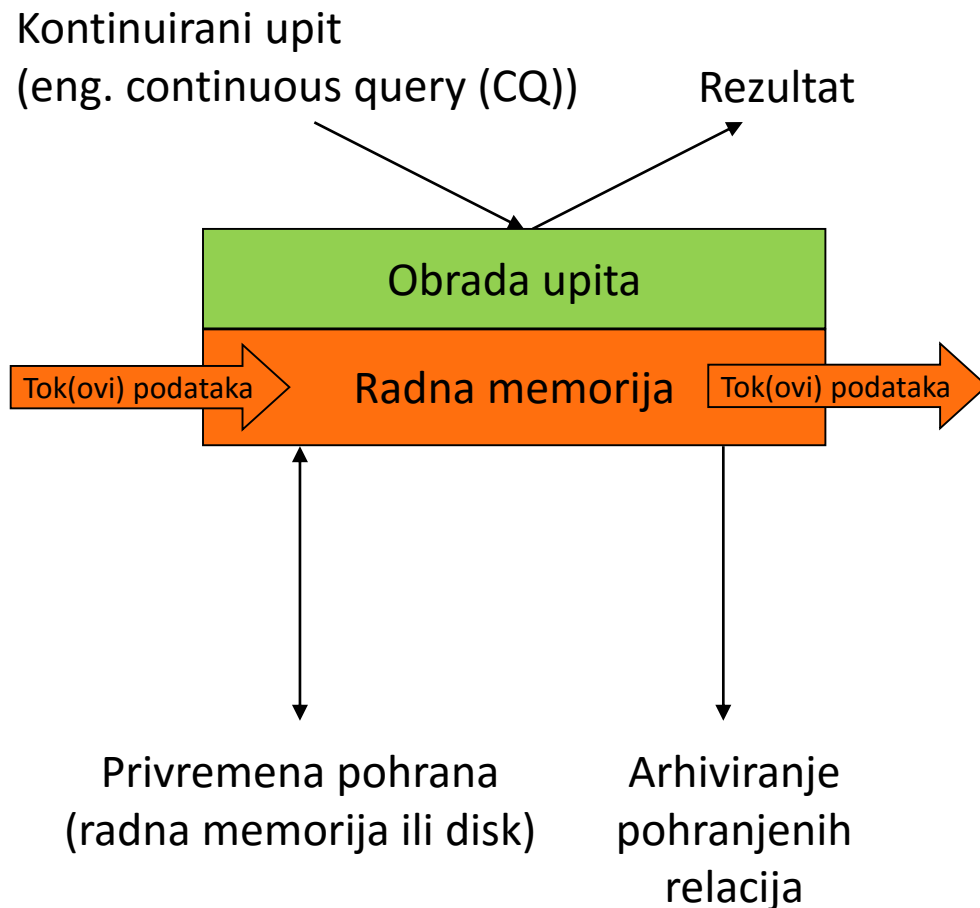
- Tradicionalni SUBP
 - **Pohranjeni** skup relativno **statičkih zapisa**
 - Dobar za aplikacije koje zahtjevaju **perzistentan spremnik podataka i kompleksne upite**
- Sustav za upravljanje tokovima podataka (SUTP)
 - Podrška za **on-line analizu** brzo mijenjajućih **tokova podataka**
 - *Tok podataka:* stvarnovremenski, uređeni slijed elemenata (implicitno prema vremenu primitka ili eksplicitno prema vremenskoj oznaci), prevelik da se pohrani u potpunosti, potencijalno neograničen
 - **Kontinuirani upiti**

SUBP vs. SUTP

Tradicionalni SUBP



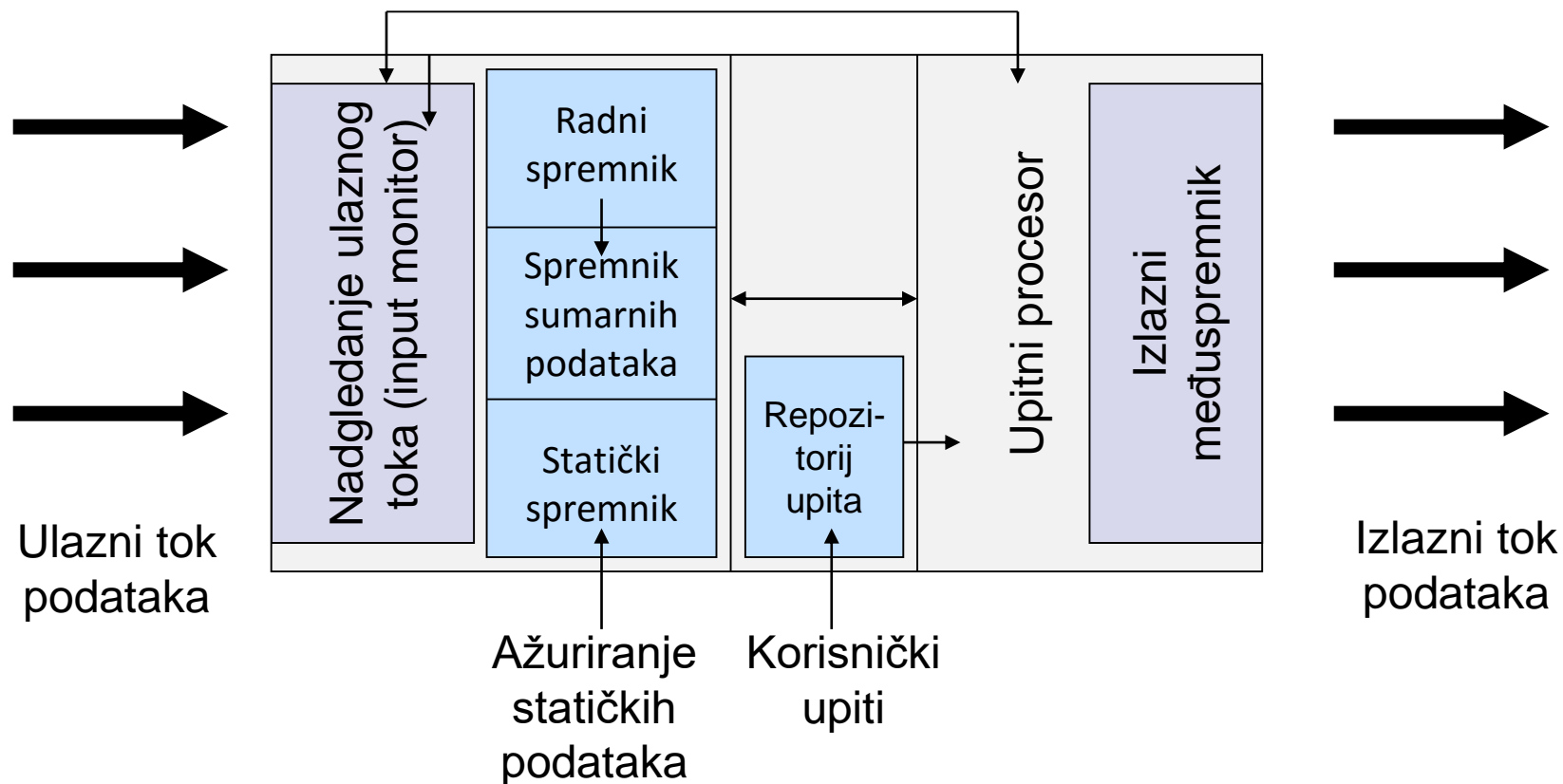
SUTP



Tradicionalni SUBP vs SUTP

Tradicionalni SUBP	SUTP
Perzistentne relacije	Tranzijentni tokovi podataka (on-line analiza)
Jednokratni upiti (one time queries)	Kontinuirani upiti (Continuous queries)
Direktan pristup podacima	Slijedni pristup podacima
“Neograničeni” spremnik za pohranu	Ograničena radna memorija
Trenutno stanje je relevantno	Bitni povijesni podaci
Nema stvarnovremenskih servisa	Stvarnovremenski zahtjevi
Relativno niska stopa ažuriranja	Potencijalno visoka stopa pritoka podataka (~GB)
Podaci bilo koje granularnosti	Podaci fine granularnosti
Očekivano precizni/točni podaci	Neprecizni/zastarjeli podaci
Pristupni plan određuje upitni procesor, dizajn BP	Nepredvidljive/varijabilne karakteristike i prtok podataka

Općenita arhitektura SUTP



Aplikacijski zahtjevi

- **Model podataka i semantika upita:** operacije temeljene na vremenu i redoslijedu
 - Selekcija
 - Ugniježdjena agregacija
 - Multipleksiranje i demultipleksiranje
 - Upiti za pronalazak frekventnih elemenata
 - Spajanja
 - Upiti temeljeni na vremenskim prozorima
- **Obrada upita:**
 - Plan izvršavanja upita mora koristiti neblokirajuće operatore
 - Algoritmi nad tokovima podataka moraju raditi **u jednom prolazu** (eng. *single-pass algorithms*)

Aplikacijski zahtjevi (2)

- **Redukcija podataka:** sumarne strukture za aproksimaciju
 - Zbornici (eng. *digests*), sažetci (eng. *synopses*) => nema egzaktnog odgovora
- **Stvarnovremenski odziv** kod nadzornih aplikacija => aktivni mehanizmi
- **Kontinuirani upiti:** varijabilna stanja sustava
- **Skalabilnost:** raspodijeljeno izvršavanje velikog broja kontinuiranih upita, nadgledanje/obrada više paralelnih tokova podataka
- **Dubinska analiza tokova podataka** (eng. *stream mining*)

Modeliranje vremena – vremenske oznake

■ Eksplicitne

- Zapisane na izvoru podataka
- U kontekstu vremenskih BP: vrijeme valjanosti (eng. *valid time*)
- Modeliraju događaj iz stvarnog svijeta reprezentiran n-torkom

■ Implicitne

- Definiraju se kao zaseban atribut definiran u okviru SUTP
- Iz perspektive SUTP – vrijeme primitka podatka (eng. *arrival time*)
- U kontekstu vremenskih BP: transakcijsko vrijeme (eng. *transaction time ili system time*)
- Omogućuju upite temeljene na redoslijedu i vremenskim prozorima

■ Problemi

- Distribuirani tokovi podataka?
- Složene (kompozitne) n-torke kreirane u okviru SUTP?

Interpunkcije

- Interpunkcije u toku podataka označavaju kraj podskupa podataka (ili razdvajaju pojedine elemente n-torke)
 - Odblokiravaju blokirajuće operatore
 - Smanjuju broj stanja za operatore ovisne o stanju
- Pomažu na dva načina
 - Održavanje redoslijeda
 - Deblokiranje sortiranja
 - Operatori koji održavaju redoslijed uključuju sort za interpunkcije
 - Dozvoljavaju narušavanje redoslijeda – razgraničavaju podskupove
 - Operatori koriste interpunkcije (a ne redoslijed!) u izlaznom rezultatu

Upiti

- SUBP: jednokratni (tranzijentni) upiti, fiksni plan izvršavanja upita optimiziran na početku
- **SUTP: kontinuirani (perzistentni) upiti, prilagodljivi upitni operatori i plan izvršavanja**
 - Podrška za perzistentne i tranzijentne upite
 - Predefinirani i ad-hoc upiti
 - Primjeri (perzistentni kontinuirani upiti)
 - Tapestry: filtriranje vijesti, e-mail poruka ... temeljno na sadržaju
 - OpenCQ, NiagaraCQ: nadgledanje web stranica
 - Chronicle: inkrementalno održavanje pogleda/virtualnih relacija (eng. *incremental view maintenance*)
- Neograničeni memorijski zahtjevi
- Blokirajući operatori: tehnike bazirane na vremenskim prozorima
- Upiti koji referenciraju povijesne podatke

Upiti (2)

- SUBP: (u većini slučajeva) egzaktan rezultat upita
- **SUTP: (u većini slučajeva) približan (aproksimativan) rezultat upita:**
 - Aproksimativni rezultati upita:
 - Izgradnja sažetaka (eng. *synopsis*): histogrami, uzorkovanje (eng. *sampling*), skice (eng. *sketches*)
 - Aproksimativni rezultat upita: koristi izgrađene sažetke
 - Aproksimativno spajanje: koristi vremenske prozore za ograničenje dosega
 - Aproksimativne agregacije: koriste izgrađene sažetke
- Grupno procesiranje podataka (eng. *batch processing*)
- Redukcija podataka: uzorkovanje, sažetci, skice, histogrami, wavelets

Upitni jezici u SUTP - paradigme

1. Relacijska

- Sintaksa bazirana na SQL-u
- Upiti referenciraju relacije ili tokove, rezultat su također relacije ili tokovi
- Napredna podrška za vremenske prozore i ordering
- Npr. StreaQuel (TelegraphCQ), AQuery, GigaScope, CQL (STREAM),

2. Objektna

- Objektno-orijentirano modeliranje toka podataka
- Klasifikacija elemenata toka na temelju hijerarhije tipova (npr. Tribeca)
- Modeliranje izvora podataka kao algebarskih tipova podataka (eng. *ADTs* – *Algebraic Data Types*) - npr. COUGAR

3. Proceduralna

- Korisnik definira (modelira) tok podataka – npr. Aurora
- Korisnik konstruira plan upita kroz grafičko korisničko sučelje
- (1) i (2) – deklarativni upitni jezici
- Trenutno najzastupljenija relacijska paradigma!

Obrada upita

1. **Neblokirajući operatori** – 3 tehnike za deblokiranje blokirajućih operatora nad tokovima podataka
 - Vremenski prozori
 - Inkrementalna evaluacija
 - Korišćenje graničnika u tokovima podataka (interpunkcije)
2. **Aproksimativni algoritmi** – ako (1) nije dovoljno, mogu se pohraniti sažeci toka podataka i nad njima izvršavati aproksimativni upiti
 - Trade-off: preciznost (accuracy) \leftrightarrow potrošnja memorije
 - Metode generiranja sažetaka: metode prebrojavanja, hash, uzorkovanje, skice, wavelet transformacije
3. **Algoritmi temeljeni na *klizajućim prozorima***
 - Uzorkovanje temeljeno na prozorima
4. **On-line dubinska analiza tokova podataka (u jednom prolazu!!)**
 - Stabla odluke, upiti temeljeni na metodi najbližeg susjeda, regresijska analiza, raspoznavanje uzoraka, ...

Primjer (TelegraphCQ)

- TelegraphCQ – baziran na PostgreSQL SUBP-u
- Tok podataka – definiran kao i svaka druga PostgreSQL tablica, koristeći PostgreSQL Data Definition Language (DDL)
 - Kreiranje toka s naredbom CREATE STREAM prije izvođenja upita
 - Podržan samo podskup SQL sintakse
 - Kontinuirani upiti – SELECT izrazi, općenita forma:

```
SELECT <select_list>
FROM <relation_and_pstream_list>
WHERE <predicate>
GROUP BY <group_by_expressions>
WINDOW stream[interval], ...
ORDER BY <order_by_expressions>;
```

- WINDOW – samo za operacije nad tokovima podataka
 - Baziran na specijalnom TIMESTAMP atributu (implicitna vremenska oznaka)

* Napomena: U ovoj ak. godini neće se tražiti na ispitu

Primjer (TelegraphCQ) – analiza mrežnog prometa

- Definicija ulaznog toka podataka – svaka n-torka sadrži:
 - polja iz zaglavlja TCP i IP paketa
 - atribut `tcqtime` za pohranu implicitno definirane vremenske oznake na ulasku n-torke u sustav
 - tip podatka `cidr` – pohrana IPv4 i IPv6 adresa u „dot” notaciji

```
CREATE STREAM network_stream (  
  ip_src cidr, ip_dst cidr, hlen bigint, tos int, length bigint, id bigint, frag_off  
  bigint, ttl bigint, prot int, ip_hcsum bigint, port_src bigint,  
  port_dst bigint, sqn bigint, ack bigint, tcp_hlen bigint, ip_len bigint,  
  flags varchar(10), window bigint, tcp_csum bigint, tcqtime  
  timestamp TIMESTAMPCOLUMN) type ARCHIVED;
```

* Napomena: U ovoj ak. godini neće se tražiti na ispitu

Primjer (TelegraphCQ) – analiza mrežnog prometa (2)

- Upit: Koliko je paketa poslano na pojedine portove u zadnjih 5 minuta?
 - Primjer spajanja između toka podataka i obične relacije (brojevi portova pohranjeni u tablici services)

```
CREATE TABLE services (port bigint, counter bigint);
```

```
SELECT services.port, count(*)  
FROM network_stream, services  
WHERE network_stream.port_dst=services.port  
GROUP BY services.port  
WINDOW network_stream ['5 min'];
```

Klizajući prozor! – svaka nova n-torka pomiče prozor, međurezultat (eng. *intermediate result*) kontinuiranog upita se vraća klijentu

* Napomena: U ovoj ak. godini neće se tražiti na ispitu

Primjer (TelegraphCQ) – analiza mrežnog prometa (3)

- Upit: Koliko podataka (bytes) je razmijenjeno na svakoj pojedinoj konekciji u zadnjoj minuti?
 - Identifikacija konekcije – paketi s istom izvornom i odredišnom IP adresom i portom pripadaju istoj konekciji (vrijedi na malom vremenskom prozoru!)

```
SELECT ip_src, port_src, ip_dst, port_dst, sum(length - ip_len - tcphlen)
FROM network_stream
GROUP BY ip_src, port_src, ip_dst, port_dst
WINDOW network_stream ['1 min'];
```

Grupiranje po konekcijama

Klijentu se vraća suma veličine podataka zapisanih u IP/TCP zaglavljima **za pojedinu grupu (konekciju)**

* Napomena: U ovoj ak. godini neće se tražiti na ispitu

Primjer (TelegraphCQ) – analiza mrežnog prometa (4)

- Upit: Koliko podataka (bytes) je razmijenjeno na svakoj pojedinoj konekciji **svakog tjedna**?
- Nemoguće riješiti kontinuiranim upitom (u trenutnoj implementaciji TelegraphCQ)
 - Problem 1:
 - GROUP BY se može koristiti samo u kombinaciji s WINDOW izrazom (prethodni primjeri)
 - Veličina prozora mora biti puno manja od „1 tjedan” jer sve podatke privremeno spremamo u radnu memoriju dok se ne obrade svi podaci iz definiranog vremenskog prozora
 - Klizajući prozor: svaki paket *više puta* doprinosi međurezultatu – za izračun ispravnog krajnjeg rezultata potrebno ukloniti redundantne podatke iz svih međurezultata
 - Moguće rješenje: nepreklapajući prozori (trenutno nisu podržani)
 - Problem 2:
 - Ista kombinacija izvorišnih i odredišnih adresa kroz veći vremenski period može pripadati različitim konekcijama - GROUP BY nije dovoljan
 - Moguće rješenje: definiramo konekciju kao „otpuštenu” (eng. *released*) ako nakon N vremenskih jedinica nema novih paketa na toj konekciji – ista kombinacija izvorišta/odredišta nakon takve stanke je nova konekcija

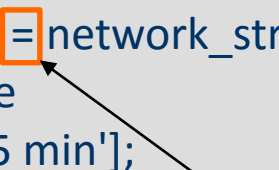
* Napomena: U ovoj ak. godini neće se tražiti na ispitu

Primjer (TelegraphCQ) – analiza mrežnog prometa (5)

- Upit: Koliko generiranog mrežnog prometa pripada pojedinim zavodima u zadnjih 5 minuta?
 - Potrebno pohraniti sve IP adrese pojedinog zavoda u zasebnu relaciju – potencijalno jako velika tablica!

```
CREATE TABLE departments (name varchar(30), ip_addr cidr, traffic bigint);
```

```
SELECT departments.name, sum(length-hlen-tcp_hlen)
FROM network_stream, departments
WHERE departments.ip_addr = network_stream.ip_src
GROUP BY departments.name
WINDOW network_stream ['5 min'];
```



Bolje rješenje:

- definirati **raspone** IP adresa za pojedini zavod
- koristiti operator „>” (provjeriti koji raspon adresa sadrži IP adresu paketa iz ulaznog toka podataka)

* Napomena: U ovoj ak. godini neće se tražiti na ispitu

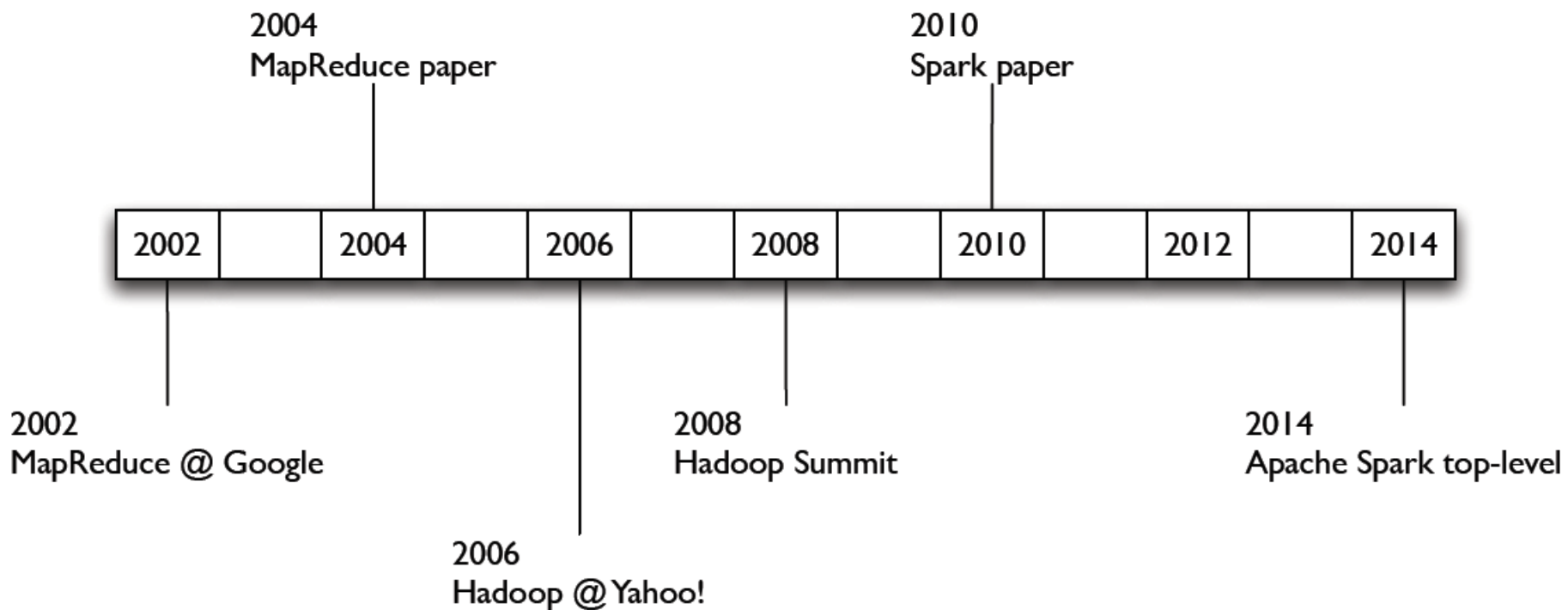
Primjeri primjene SUTP

- **Senzorske mreže**
 - Nadgledanje senzorskih podataka s velikog broja izvora, kompleksno filtriranje, aktivacija alarma, agregacije i spajanja nad jednim ili više tokova podataka
- **Analiza mrežnog prometa**
 - Analiza Internet prometa u stvarnom vremenu – izračun statističkih podataka i detekcija kritičnih stanja
- **Financijski podaci**
 - On-line analiza cijena dionica, valuta (npr. BitCoin), otkrivanje korelacija, identifikacija trendova
- **On-line aukcije**
- **Analiza transakcijskih dnevnika** (npr. Web, telefonski pozivi, ...)

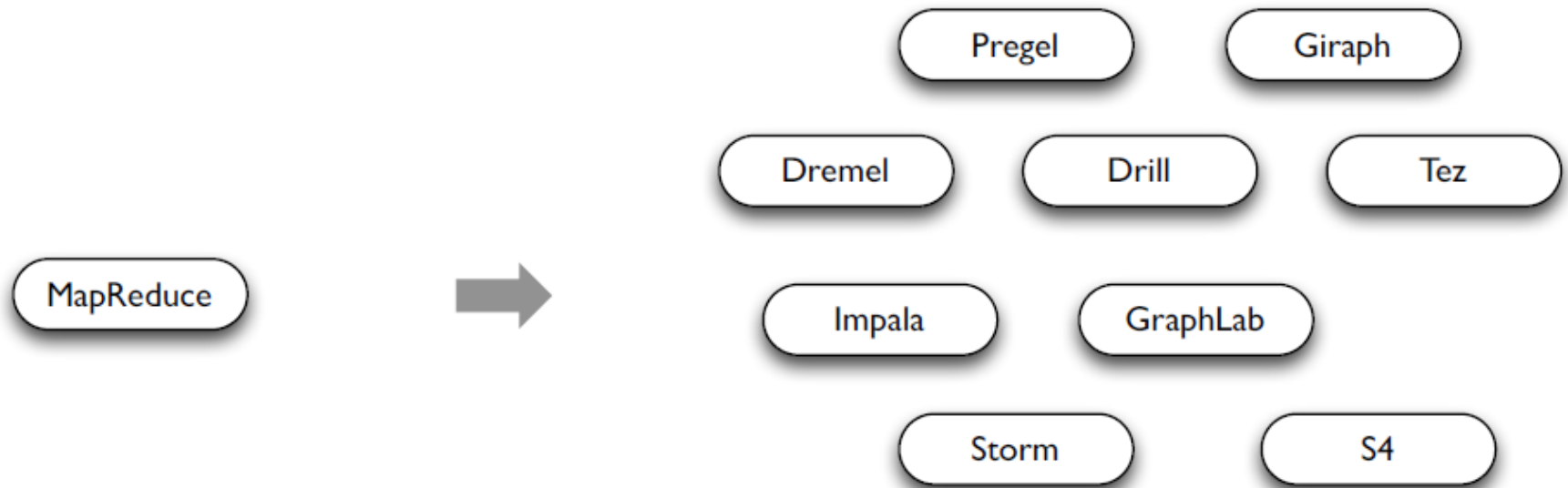
Distribuirana obrada tokova podataka

Apache Spark Streaming

Povijest ukratko



Povijest ukratko – MapReduce (MR)



General Batch Processing

Specialized Systems:

iterative, interactive, streaming, graph, etc.

Povijest ukratko – Apache Spark

- Razvijen 2009.g. na UC Berkley AMPLab
- 2010.g. – open-source!
- 2016.g. – jedna od najvećih community-ja u BigData području
 - Više od 200 kontributora u 50+ organizacija

“Organizations that are looking at big data challenges – including collection, ETL, storage, exploration and analytics – should consider Spark for its in-memory performance and the breadth of its model. It supports advanced analytics solutions on Hadoop clusters, including the iterative model required for machine learning and graph analysis.”

Gartner, Advanced Analytics and Data Science (2014)

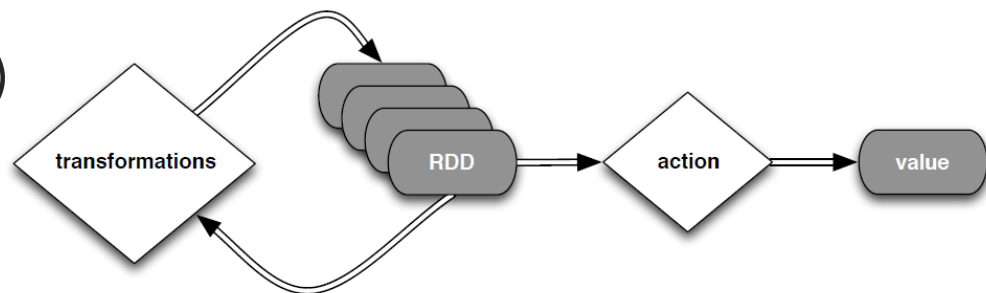


Apache Spark – osnovne karakteristike

- Obrada skupnih, interaktivnih i stvarnovremenskih podataka na temelju istog programskog okvira
- Ugrađena podrška za integraciju:
 - Java, Python, Scala
- Razvoj aplikacija na višoj razini apstrakcije
- Općenitiji pristup
 - MapReduce je samo jedan od dostupnih konstrukata!
 - Unificirana platforma za brojne različite slučajeve korištenja
- Lijena evaluacija (eng. *lazy evaluation*)
 - Redukcija čekajućih stanja, bolji pipeline
- Funkcijsko programiranje / jednostavnost korištenja
 - Redukcija troška održavanja velikih aplikacija/sustava

Resilient Distributed Datasets

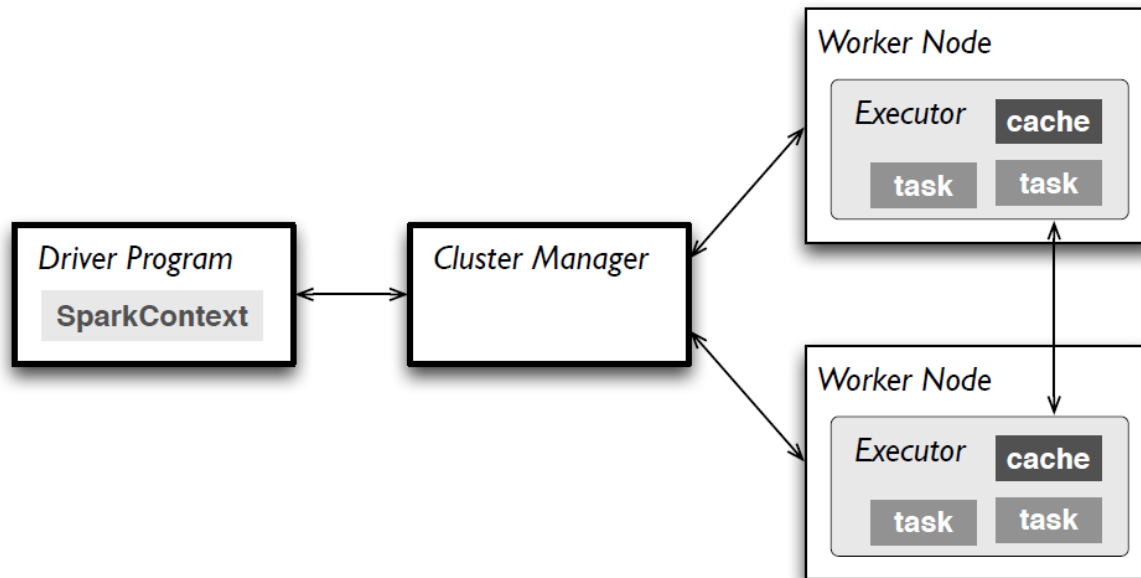
- **Resilient Distributed Datasets (RDD)**
 - Primarna apstrakcija podataka u Spark-u
 - Kolekcija elemenata otporna na pogreške, koja se može obrađivati u **paraleli**
 - Moguće perzistirati u radnoj memoriji ili na disk
- Trenutno dvije vrste:
 - *Paralelizirane kolekcije* – koristi postojeće Scala kolekcije i izvršava nad njima funkcije u paraleli
 - *Hadoop skupovi podataka* – izvršavanje funkcija nad svakim zapisom datoteke temeljene na Hadoop distribuiranom datotečnom sustavu (ili bilo kojem drugom sustavu pohrane koji je podržan kroz Hadoop platformu)
- Dvije vrste operacija:
 - Transformacije (lijena evaluacija)
 - Akcije
 - Transformirani RDD se evaluira kad se nad njim izvrši akcija!



Spark programi - *SparkContext*

- Svaki Spark program (aplikacija) – kao prvi korak kreira SparkContext objekt
 - Definira kako Spark pristupa raspodijeljenom skupu čvorova za obradu podataka (cluster)
 - Parametar master definira cluster koji se koristi
 - Kroz Spark ljusku – varijabla sc
 - Samostojeći programi – instanciraju novi SparkContext objekt
 - Koristi se za kreiranje drugih varijabli u programu

Spark – distribuirana obrada



1. Master se spaja na *cluster manager* i alocira potrebne resurse
2. Pokreće izvršitelje (eng. *executors*) na čvorovima clustera – procesi izvršavaju obradu, spremaju podatke u priručnu memoriju (eng. *caching*)
3. Master šalje aplikacijski kod izvršiteljima
4. Master šalje zadatke (eng. *tasks*) izvršiteljima na obradu

Apache Spark - komponente

BlinkDB
*Approximate
SQL*

Spark SQL
SQL

Spark
Streaming
Streaming

MLlib
*Machine
Learning*

GraphX
*Graph
Computation*

Spark R
R on Spark

Spark Core Engine

Alpha / Pre-alpha

Spark Streaming

- Omogućuje jednostavnu izgradnju skalabilnih aplikacija, otpornih na pogreške (eng. *fault-tolerant*), baziranih na **stvarnovremenskim tokovima podataka**



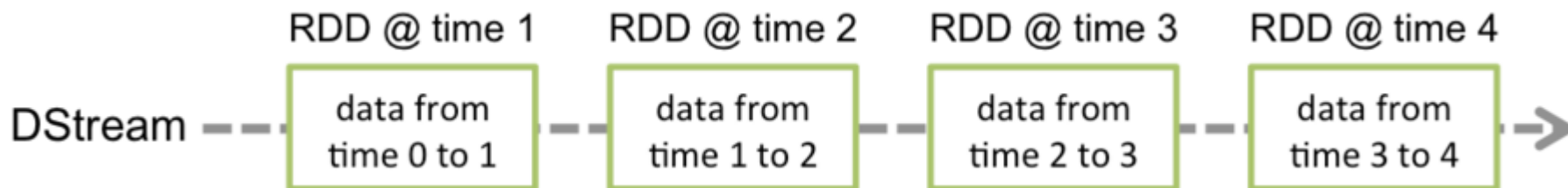
- Obrada podataka:

1. Zaprimanje stvarnovremenskog ulaznog toka podataka
2. Podjela ulaznog toka u grupe (eng. *batches*)
3. Procesiranje pojedinih grupa podataka kroz jezgrieni Spark sustav
4. Generiranje izlaznog toka (grupirani podaci)



Spark Streaming - *DStream*

- Diskretizirani tok podataka (eng. *discretized stream*) – **DStream**
 - Apstrakcija podataka visoke razine
 - Reprezentacija kontinuiranog toka podataka
- Može biti kreiran iz:
 - Ulaznog toka podataka s nekog od vanjskih izvora (npr. Twitter)
 - Primjenom operacija (transformacija) nad tokom podataka
- Interno: reprezentiran kao slijed RDD objekata



Literatura

- Charu C. Aggarwal, *Data streams: models and algorithms*, Springer Science+Business Media, LLC, 2007.g.
- S. Muthukrishnan, *Data streams: algorithms and applications*, Foundations and Trends in Theoretical Computer Science Vol. 1 Issue 2, August 2005
- Vera Goebel, *Data Stream Management Systems*, „Advanced Database Systems” course slides, University of Oslo, Department of Informatics, 2015.g., <http://www.uio.no/studier/emner/matnat/ifi/INF5100/index-eng.html>
- Lukasz Golab, M. Tamer Özsu, *Issues in data stream management*, ACM SIGMOD Record, Vol. 32, Issue 2, 2003.g.
- Plagemann T. et al, *Using Data Stream Management Systems for Traffic Analysis – A Case Study*, Volume 3015 of the series Lecture Notes in Computer Science pp 215-226, SpringerLink, 2004.g.
- Apache Spark documentation, <http://spark.apache.org/docs/latest/>, 2015.g.