Influx - Time Series DB

NoSQL baze podataka

Univerzitet u Novom Sadu
Fakultet tehničkih nauka

Influx baza podataka

- *Zvanična dokumentacija*
- *Go driver*

Influx baza podataka

- Open-source
- Napisana u GoLang-u
- Namenski kreirana za skladištenje i rukovanje velikom količinom vremenskih podataka
- Slična struktura podataka kao *Cassandra*
- Horizontalno skaliranje kroz *clustering*
- Visoka dostupnost kroz *replikacije*
- Preciznost do nivoa nanosekunde
- Raznovrsna podrška za pretrage i agregacije = **Flux upitni jezik**

Osnovni pojmovi

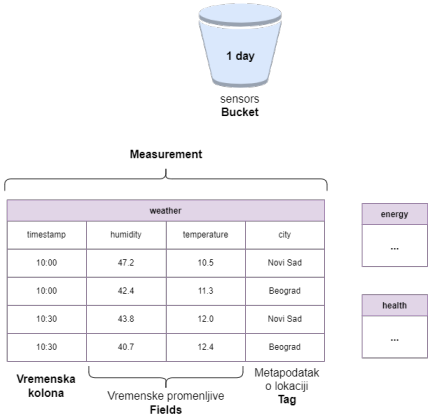
Uporedni pregled pojmova u okviru InfluxDB i SQL baza podataka:

- Bucket
- Measurement
- Field - vremenska promenljiva koju pratimo
- Tag - metapodaci o vremenskog promenljivoj
- Point (tačka) - vremenski momenat sa vrednostima vremenskih promenljivih

- Baza podataka
- Tabela
- Kolona
- Indeksirana kolona
- Torka

Napomena: mapiranje nije 1-1!

Primer



- Zašto su *humidity* i *temperature* fields, a *city* tag?

Podrška

- Čuvanje podataka na osnovu vremena
- Kompresija podataka na osnovu vremena
- Rukovanje *životnim ciklusom podataka*
  - Senzor očita vrednost svaku sekundu *86 400* zapisa u toku dana
  - Nagomilavanje podataka velike preciznosti
  - Kratak vremenski period čuvamo izuzetno tačne podatke
  - Nakon nekog vremena (na dnevnom, mesečnom, godišnjem) nivou, sažimamo podatke u manje precizne
  - **Retention policy** - politika čuvanja podataka na nivou bucket-a (koliko dugo čuvamo podatke)
  - *Primer: temperatura*
- Sumarizacija podataka
- Dobavljanje velike količine podataka za vremenski opseg
- Vremenski svesni upiti (sve što se desilo u prethodnih 12 sati od sadašnjeg trenutka)

Primena

- Meteorološki podaci
- IoT senzori (medicina, industrija)
- Podaci o proizvodnji i prodaji
- Praćenje tržišta
- Praćenje akcija, kripto valuta, finansije...
- Bilo šta što zavisi od jedinice vremena

Skladištenje - kako?

- **TSM** = Time Structured Merge Tree
- Posebna struktura podataka kreirana za rad sa vremenskim podacima
- Uzor: **LSM** = Log Structured Merge Tree
- LSM koriste *Cassandra, HBase,...*
- Zašto LSM kao uzor?
- Više o strukturama na:
  - *LSM naučni rad*,
  - *LSM objašnjenje*,
  - *TSM*

Skladištenje - zašto?

- Dobre perofrmanse prilikom čitanja
- Bolje performanse prilikom pisanja podataka
- Slabije performanse prilikom izmene i brisanja
- Da li menjamo vremenske podatke?

Skladištenje - struktura

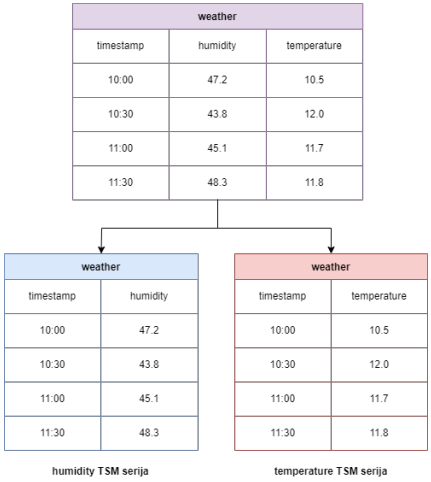
- **TSM serija** = promena **jedne** mere u jedinici vremena
- Ako pratimo više mera (temperatura, vlažnost, jačina vetra, smer vetra,...), za svaku meru kreira se posebna **TSM serija**
- **Field key = mera**
- **Field value = vrednost mere**
- **Tag = metapodatak vezan za meru**
- Svaka jedinstvena vrednost tag kolone predstavlja posebnu grupu
- Za svaku grupu kreira se **TSM serija**
- Svaka jedinstvena kombinacija vrednosti **field key** i **tag value** predstavlja **1 TSM seriju**
- **Kardinalitet** = ukupan broj TSM serija
- Visok kardinalitet može dovesti do pada u performansama!

Skladištenje - primer fields

weather		
timestamp	humidity	temperature
10:00	47.2	10.5
10:30	43.8	12.0
11:00	45.1	11.7
11:30	48.3	11.8

- Koji je kardinalitet podataka?

Skladištenje - primer fields rešenje

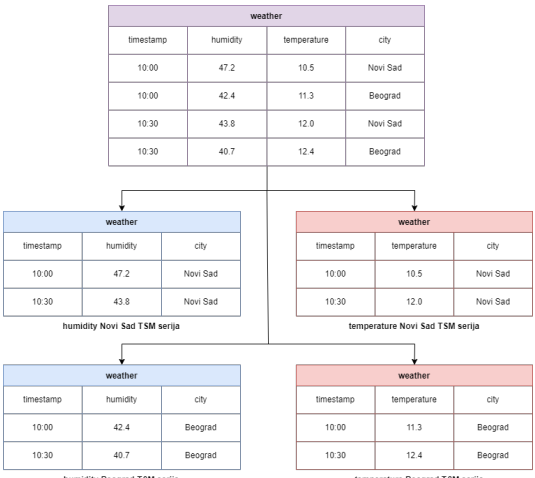


Skladištenje - primer tags

weather			
timestamp	humidity	temperature	city
10:00	47.2	10.5	Novi Sad
10:00	42.4	11.3	Beograd
10:30	43.8	12.0	Novi Sad
10:30	40.7	12.4	Beograd

- Koji je kardinalitet podataka?
- Kako biramo da li je kolona *tag* ili *field*?

Skladištenje - primer tags rešenje



Modelovanje podataka

- Da li podatak odgovara tag-u ili field-u?
- Na koji način dobavljam podatke?
- Poznavanje upita nad podacima unapred (slično Cassandri)
- Problem visokog kardinalitet
- Tag key i field key ne treba da sadrže podatke već da budu intuitivni (slično imenu kolone)

Line protokol

- Tekstualni format zapisa vremenskih tačaka
- Case-sensitive
- **White-space sensitive** - svaka belina se mora eskejpovati!
- Sintaksa:

```
ime_measurementa,tag_key1=tag_value1,tag_key2=tag_value2
↪  field_key1=field_value1,field_key2=field_value2 unix_format_timestamp
```

- Primer:

```
vreme,lokacija=Novi\ Sad temperatura=27,smerVetra="SZ" 1670866214
```

- Više na: *Line protokol*

Modelovanje podataka - primeri

- Dobar model

```
weather_sensor,crop=blueberries,plot=1,region=north temp=50.1 1472515200000000000
weather_sensor,crop=blueberries,plot=2,region=midwest temp=49.8 1472515200000000000
```

- Loš model

```
blueberries.plot-1.north temp=50.1 1472515200000000000
blueberries.plot-2.midwest temp=49.8 1472515200000000000
```

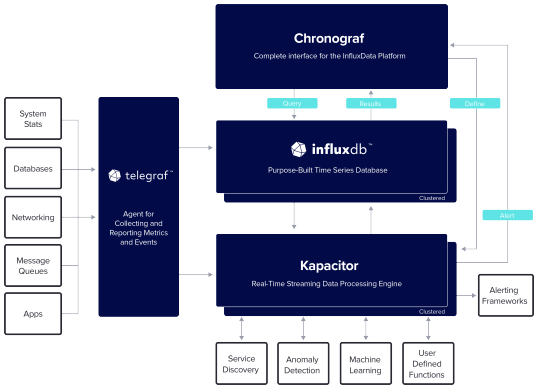
Tipovi podataka

- Vrednost field-a može biti:
  - Integer
  - Float
  - String
  - Boolean
- Vrednost tag-a može biti:
  - String

TICK

- Kolekcija open-source alata za rad sa time seris podacima
- **T**elegraf - prikupljanje podataka
- InfluxDB - skladištenje podataka
- **C**hronograf - vizuelizacija podataka
- **K**apacitor - procesiranje i monitoring podataka (uz alerting sistem)

TICK stack


Chronograf
Complete interface for the InfluxData Platform
Query
Results
Define
System Stats
Databases
Networking
Message Queues
Apps
telegraf
Agent for Collecting and Reporting Metrics and Events
influxdb
Purpose-Built Time Series Database
Clustered
Alert
Kapacitor
Real-Time Streaming Data Processing Engine
Clustered
Alerting Frameworks
Service Discovery
Anomaly Detection
Machine Learning
User Defined Functions

flux

- Nije klasičan CLI alat, svaka komanda se izvršava posebno
- **docker exec -it ime_kontejnera bash** - povezivanje na *bash* u okviru pokrenutog Docker kontejnera

Create

- **influx write** - upis jedne tačke čiji zapis prati *Line protokol*
- **influx API** - pozivom write metode influx API-ja
- **UI upload** - upload fajlova u *annotated CSV* ili line protokolu
- UI upload se svodi na poziv influx write uz podršku za čitanje .csv fajlova

Read

1. **from bucket_name** - definisanje izora podataka
2. **range(start: start, stop: stop)** - definisanje vremenskog intervala (start je obavezan parametar, stop default-no uzima trenutno vreme)
3. **filter(fn: function)** - filtriranje podataka; *function* je predicate funkcija koja definiše krijterijum(e) filtriranja
4. **yield** - vraćanje vrednosti izraza; neophodno navesti samo u slučaju više vezanih upita

Read - transformacije

- **agregacije** - operacije nad podacima
- **downsampling** - sažimanje podataka
- **mean, max, min, meadian, group,...** - podržane funkcije za rad sa podacima
- **window** - particionisanje podataka na osnovu vremena

Delete

- **influx delete** - brisanje je moguće na nivou *vremenskog intervala*, *vrednosti tag-a*, *vrednosti field-a* i na nivou celog *measurement-a*

Update

- Izmena vrednosti field-a <=> **upisivanju nove tačke** koja će se razlikovati samo u vrednosti field-a
- Izmena vrednosti tag-a <=> **brisanje tačke** i upisivanje nove sa željenim vrednostima
- Rad sa duplikatima: *link*

Primer

U okviru primera *RestInflux* upotrebljeni su:

1. *InfluxDB* - baza podataka
2. *data.txt* - skripta za popunu baze test podacima
3. *Docker* - kontejnerizacija rešenja (i "instalacija" baza)
4. *Go* - implementacija primera

Zadaci

- Napisati komandu za prikaz prosečne temperature na godišnjem nivou
- Napisati komandu za prikaz minimalne i maksimalne temperature na godišnjem nivou
- Napisati komandu za prikaz prosečne temperature na kvartalnom nivou (3 meseca)
- Napisati komandu za prikaz prosečne temperature u Torontu
- Napisati komandu za prikaz prosečne temperature u Srbiji
- **Bonus**: Smisliti i isprobati upite po izboru