Indexing

CO TO JEST

I jak działa?

RDF RIOTECH DATA FACTORY

Indexing

- Proces zapisu do Solr
- Podczas indeksacji, Solr uwzględnia analizę danych (analyzers)
- Czym jest “inverted index”?

Inverted Index

ID: 1
Description:
“Ala ma kota i psa”

ID: 2
Description:
“Kot i pies byli na zakupach”

ID: 3
Description:
“Nasze zakupy to być lub nie być małych sklepów”

Analiza + indexing

ID: 1
Description:
“Ala nie ma kota i psa”

ID: 2
Description:
“Kot i pies byli na zakupach”

ID: 3
Description:
“Nasze zakupy to
być lub nie być
małych sklepów”

RDF RIOTECH DATA FACTORY

Analiza + indexing

ID: 1
Description:
“Ala nie ma kota i psa”

ID: 2
Description:
“Kot i pies byli na zakupach”

ID: 3
Description:
“Nasze zakupy to być lub nie być małych sklepów”

Term	Documents
ala	1
kot	1,2
pies	1,2
zakupy	2,3
być	2,3
nasze	3
nie	1,3
małych	3
sklep	3

Indexing

Jak załadować dane?

- Upload pliku XML/csv/json przez wysyłkę HTTP do Solr.
  - Podany wprost dokument
  - Plik z wieloma dokumentami
- Użycie Java API i napisanie swojego indexera (lub pokrewne: wykorzystanie Apache Spark)
- ~~Użycie Solr Cell~~ (niezalecane, przestarzałe)
- Narzędzie bin/post (do celów developerskich)

Indexing

Pola

- Dokument musi mieć pola zgodne ze schematem (managed-schema)

- Jeśli pola nie występują w schemie, Solr zignoruje je lub przypisze do zmiennych dynamicznych,

- Należy podać unikatowe pole (zwykle ID). Jeśli nie podamy, Solr sam je wygeneruje.

Proces indexingu

- Dokumenty są przyjmowane i przekazywane konkretnym liderom.
- Z lidera wysyłane są do replik.
- Replika odpowiada liderowi, a lider pierwotnemu nodowi.
- Gdy liderzy odpowiedzą, oryginalny node odpowiada klientowi. Dokumenty są zapisywane do T-Logów (Transaction Logs)
- W T-Logach są przechowywane do czasu commita, który przerzuci je do indexu.

Proces indexingu

commits

- Hard commit - przerywa t-log i rozpoczyna następny (zasobożerne zdarzenie)
- Soft commit - pozwala przekazywać dane do indexu bez przerywania t-loga.
  - Mniej zasobożerne, ale nie darmowe.
  - Wymaga zasobów i unieważnia wysokopoziomowe cache.

Near Real Time

- Tradycyjnie dane nie są indexowane od razu po przesłaniu, ale dopiero po jakimś czasie (lub przy określonych wielkościach)
- Po co? Żeby nie uruchamiać kosztownego hard-commita przy każdym dorzuceniu do pieca.
- Pytanie: jak sprawić, aby dane były dostępne automatycznie?
- Odpowiedź: Near Real Time, czyli:
  - Dodanie soft commit
  - GET API

Near Real Time

Konfiguracja soft commit

```
<autoSoftCommit>

  <maxTime>6000</maxTime>

  <maxDocs>1000</maxDocs>

</autoSoftCommit>
```

- maxTime - w milisekundach od ostatniego niezakomitowanego updatu
- Całość ustawiamy w solrconfig.xml

XML, CSV, HTTP

RDF RIOTECH DATA FACTORY

Indeksowanie XML

- Możemy indexować dane podając różne formaty danych: json, xml, csv itd.

- XML to struktura, którą możemy wykorzystać do zapisania danych, które wylądują w Solr.

- Możemy je zarówno przekazywać wprost (przy okazji requestów), jak i zapisywać w pliku i adres do pliku przekazywać do requesta.

- Można przesyłać dane XMLowo zarówno poprzez narzędzie bin/post, jak i HTTP.

Indeksowanie XML

Struktura

- <add> - wprowadza jeden lub więcej dokumentów do dodania.
- <delete> - wprowadza jeden lub więcej dokuemntów do usunięcia, poprzez podanie id lub przekazując query
  - <query>price:12</query>
- <doc> - Opisuje dokument
- <field> - Opisuje pojedyncze pole
- W <add> może być wiele dokumentów.

Indeksowanie XML

Struktura

```
<add>
	<doc>
		<field name="id">1</field>
		<field name="title">25 hours a day</field>
		<field name="author">Nick Bare</field>
		<field name="price">43</field>
		<field name="description">You have big dreams that fire you up, and yet a fear of failure is holding you back. You see the success others have achieved and
		doubt you could ever do what they’ve done. You tell yourself you lack the smarts, skills, or leadership capabilities to live out your dream, but the truth is,
		there’s a massive gap between what you think you can do and what you’re actually capable of doing. Nick Bare wants to help you close that gap.

		In Twenty-Five Hours A Day, Nick shares the lessons he learned while building his business as a member of the US Army. He grew that business to seven figures
		by flipping the switch and going all in—then breaking that switch so he could never go back. Now, Nick wants to help you transform your life by embracing the
		suck, living like you’ve got an extra hour, and harnessing the power of “one more.” By applying these simple lessons, you’ll radically improve your chances of
		success.</field>
		<field name="category">Biznes</field>
	</doc>
	<doc>
		<field name="id">2</field>
		<field name="title">Hooked: How to Build Habit-Forming Products</field>
		<field name="author">Nir Eyal</field>
		<field name="price">59</field>
		<field name="description">Nir Eyal reveals how successful companies create products people can't put down - and how you can too

		Why do some products capture our attention while others flop? What makes us engage with certain things out of sheer habit? Is there an underlying pattern to
		how technologies hook us?

		Nir Eyal answers these questions (and many more) with the Hook Model - a four-step process that, when embedded into products, subtly encourages customer
		behaviour. Through consecutive "hook cycles," these products bring people back again and again without depending on costly advertising or aggressive messaging.

		Hooked is based on Eyal's years of research, consulting, and practical experience. He wrote the book he wished had been available to him as a start-up founder
		- not abstract theory, but a how-to guide for building better products. Hooked is written for product managers, designers, marketers, start-up founders, and
		anyone who seeks to understand how products influence our behaviour.

		Eyal provides readers with practical insights to create user habits that stick; actionable steps for building products people love; and riveting examples from
		the iPhone to Twitter, Pinterest and the Bible App.</field>
		<field name="category">Biznes</field>
	</doc>
</add>
```

Indeksowanie XML

Struktura

<update> - pozwala zgrupować operacje i dodać oraz usunąć wiele różnych dokumentów

```
<update>
        <add>
                <doc>
                        <field name="id">1</field>
                        <field name="title">25 hours a day</field>
                        <field name="author">Nick Bare</field>
                        <field name="price">43</field>
                        <field name="description">You have big dreams that fire you up, and yet a fear of failure is holding you back. You see the success others have achieved
                        and doubt you could ever do what they've done. You tell yourself you lack the smarts, skills, or leadership capabilities to live out your dream, but
                        the truth is, there's a massive gap between what you think you can do and what you're actually capable of doing. Nick Bare wants to help you close that
                        gap.

                        In Twenty-Five Hours A Day, Nick shares the lessons he learned while building his business as a member of the US Army. He grew that business to seven
                        figures by flipping the switch and going all in—then breaking that switch so he could never go back. Now, Nick wants to help you transform your life by
                        embracing the suck, living like you've got an extra hour, and harnessing the power of "one more." By applying these simple lessons, you'll radically
                        improve your chances of success.</field>
                        <field name="category">Biznes</field>
                </doc>
        </add>
        <delete>
                <id>1234</id>
        </delete>
</update>
```

Indeksowanie XML

Request HTTP

```
curl http://localhost:8983/solr/my_collection/update -H "Content-Type: text/xml" --data-binary '
<add>
  <doc>
    <field name="authors">Patrick Eagar</field>
    <field name="subject">Sports</field>
    <field name="dd">796.35</field>
    <field name="isbn">0002166313</field>
    <field name="yearpub">1982</field>
    <field name="publisher">Collins</field>
  </doc>
</add>'
```

Indeksowanie XML

Request HTTP

```
curl http://headnode.rdf.com:8983/solr/books3/update -H "Content-Type: text/xml" --data-binary @books.xml
```

```
curl http://headnode.rdf.com:8983/solr/books3/update -H "Content-Type: text/xml" -T "books.xml" -X POST
```

```
maro@headnode:~/solr/data$ curl http://h
<?xml version="1.0" encoding="UTF-8"?>
<response>

<lst name="responseHeader">
  <int name="rf">1</int>
  <int name="status">0</int>
  <int name="QTime">374</int>
</lst>
</response>
maro@headnode:~/solr/data$
```

Indeksowanie XML

Request HTTP - commit

```
root@06ee18501a8c:~/solr/data# curl http://localhost:8983/solr/recipes/update?commit=true
{
  "responseHeader":{
    "status":0,
    "QTime":21}}
```

Indeksowanie XML

Narzędzie post

```
/bin/post -c books3 ~/solr/data/books-simple.xml
```

Atomic Update

- Mechanizm pozwalający na update części dokumentu, bez potrzeby przeindeksowania całego.

- Całość odbywa się tak samo jak zwykła indeksacja całego dokumentu

- Podczas dodawania pola, możemy użyć kilku modyfikatorów, które stosujemy do pól.

Atomic Update

Modyfikatory

- set - wstawia lub wymienia wartości w danym polu. Można także wstawić “null”, aby wymienić wartości.

- add - dodaje wartość do listy (pola multiValued). Można określić pojedynczą wartość lub listę.

- add-distinct - Dodaje określone wartości, ale tylko jeśli takowe jeszcze nie istnieją.

- remove - wywala wszystkie wystąpienia danej wartości w liście (pole multiValued).

- removeregex - wyrzuca wszystkie pasujące do wzorca regex.

- inc - podnosi wartość liczbową o określoną liczbę

Atomic Update

Przykład

```
curl http://headnode.bda.rdf.com:8983/solr/books_test/
update -H "Content-Type: text/xml" --data-binary '
<add>
  <doc>
    <field name="id">3</field>
    <field name="price" update = "set">90</field>
  </doc>
</add>’
```

Atomic Update

Przykład


```
{
  "id":"3",
  "title":"Allgäu 2: Ostallgäu und vorderes Lechtal 55 Touren mit GPS-Tracks",
  "suggest_field":["Allgäu 2: Ostallgäu und vorderes Lechtal 55 Touren mit GPS-Tracks"],
  "author":"Mark Zahel",
  "price___s_ns":["PLN",
    "PLN"],
  "price":"90,PLN",
  "description_german":["Die Schlösser Neuschwanstein und Hohenschwangau, Bergseen und Weiher,
  "title_german":["Allgäu 2: Ostallgäu und vorderes Lechtal 55 Touren mit GPS-Tracks"],
  "_version_":1758597761681326080}]
```

Gratulacje!

Good job;-)

RDF RIOTECH DATA FACTORY