



Apache Solr

Востриков Даниил, 02.05.2022

История развития Apache Solr

Йоник Сили создал Solr в 2004 году, чтобы добавить возможности поиска на веб-сайт компании CNET Networks. В январе 2006 года был сделан проект с открытым исходным кодом в рамках Apache Software Foundation. Его последняя версия, Solr 6.0, была выпущена в 2016 году с поддержкой выполнения параллельных SQL-запросов.

Solr можно использовать вместе с Hadoop. Поскольку Hadoop обрабатывает большой объем данных, Solr помогает нам найти необходимую информацию из такого большого источника. Solr может использоваться не только для поиска, но и для хранения. Как и другие базы данных NoSQL, это нереляционная технология хранения и обработки данных .

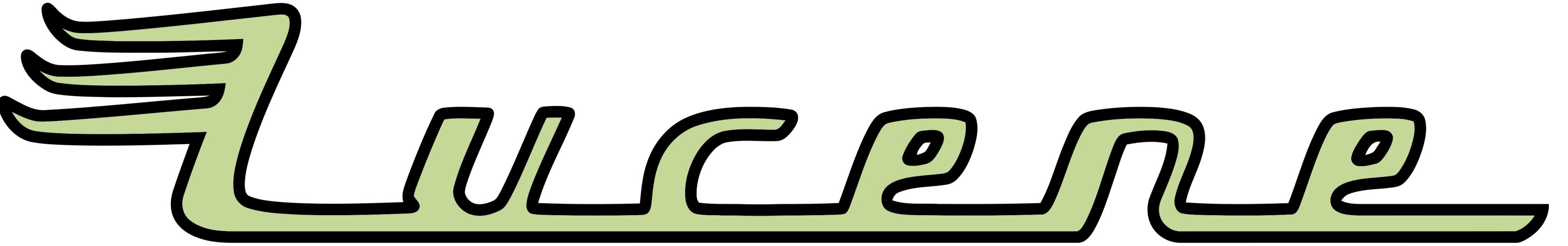
Короче говоря, Solr – это масштабируемая, готовая к развертыванию система поиска и хранения, оптимизированная для поиска больших объемов тексто-ориентированных данных.

Особенности Apache Solr

- **Restful APIs** - чтобы общаться с Solr, не обязательно иметь навыки программирования на Java. Вместо этого вы можете использовать успокоительные услуги для общения с ним. Мы вводим документы в Solr в таких форматах, как XML, JSON и .CSV, и получаем результаты в тех же форматах.
- **Полнотекстовый поиск** - Solr предоставляет все возможности, необходимые для полнотекстового поиска, такие как токены, фразы, проверка орфографии, подстановочные знаки и автозаполнение.
- **Готовность для предприятия** - в зависимости от потребностей организации Solr может быть развернут в любых системах (больших или малых), таких как автономные, распределенные, облачные и т. Д.
- **Гибкость и расширяемость** - расширяя классы Java и настраивая их соответствующим образом, мы можем легко настраивать компоненты Solr.
- **База данных NoSQL** - Solr также может использоваться в качестве базы данных NOSQL с большими объемами данных, где мы можем распределять задачи поиска по кластеру.
- **Интерфейс администратора** - Solr предоставляет простой в использовании, удобный, функциональный пользовательский интерфейс, с помощью которого мы можем выполнять все возможные задачи, такие как управление журналами, добавление, удаление, обновление и поиск документов.
- **Высокая масштабируемость** - используя Solr с Hadoop, мы можем масштабировать его емкость, добавляя реплики.
- **Текст-ориентированный и отсортированный по релевантности** - Solr в основном используется для поиска текстовых документов, а результаты предоставляются в соответствии с запросом пользователя по порядку.

Взаимодействие с Apache Solr

- Из коробки: <https://solr.apache.org/downloads.html>
- Docker: <https://github.com/docker-solr/docker-solr#getting-started-with-the-docker-image>
- Kubernetes: <https://solr.apache.org/operator/resources.html#tutorials>



- Solr – это оболочка вокруг Java API от Lucene. Поэтому, используя Solr, вы можете использовать все возможности Lucene.
- Lucene – это простая, но мощная библиотека поиска на основе Java. Ее можно использовать в любом приложении для добавления возможностей поиска. Библиотека Lucene предоставляет основные операции, необходимые для любого поискового приложения, такие как **индексирование и поиск**.
- В отличие от Lucene, вам не нужно иметь навыки программирования на Java при работе с Apache Solr. Он предоставляет замечательную готовую к развертыванию службу для создания окна поиска с автозаполнением, которое Lucene не предоставляет.

Язык запросов в Apache Solr

- После установки solr переходим в папку **bin** в домашнем каталоге solr и запускаем solr, используя следующую команду.
- Эта команда запускает solr в фоновом режиме, прослушивая порт 8983, отображая следующее сообщение.

```
(base) MacBook-Pro-Daniil-2:solr-8.11.1 daniilvostrikov$ bin/solr start
*** [WARN] *** Your open file limit is currently 2560.
It should be set to 65000 to avoid operational disruption.
If you no longer wish to see this warning, set SOLR_ULIMIT_CHECKS to false in your profile or solr.in.sh
*** [WARN] *** Your Max Processes Limit is currently 1392.
It should be set to 65000 to avoid operational disruption.
If you no longer wish to see this warning, set SOLR_ULIMIT_CHECKS to false in your profile or solr.in.sh
Waiting up to 180 seconds to see Solr running on port 8983 [-]
Started Solr server on port 8983 (pid=24426). Happy searching!
```

Язык запросов в Apache Solr

Далее работать будем в интерфейсе Solr Admin, используя следующий адрес:

<http://localhost:8983/solr/>

The screenshot shows the Apache Solr Admin interface at <http://localhost:8983/solr/>. The dashboard displays various metrics and configuration details. On the left, a sidebar menu includes links for Dashboard, Logging, Security, Core Admin, Java Properties, Thread Dump, and a note about No cores available. The main content area is divided into several sections: **Instance** (Start button, last updated 'about a minute ago'), **Versions** (solr-spec 8.11.1, solr-impl 8.11.1, lucene-spec 8.11.1, lucene-impl 8.11.1), **JVM** (Runtime Oracle Corporation Java HotSpot(TM) 64-Bit Server VM 17.0.1 17.0.1+12-LTS-39, Processors 8, Args -DSTOP.KEY=solrrocks, -DSTOP.PORT=7983, -Djetty.home=/Users/daniilvostrikov/Desktop/solr-8.11.1/server, -Djetty.port=8983, -Dsolr.data.home=, -Dsolr.default.confdir=/Users/daniilvostrikov/Desktop/solr-8.11.1/server/solr/configse, -Dsolr.install.dir=/Users/daniilvostrikov/Desktop/solr-8.11.1, -Dsolr.jetty.inetaccess.excludes=, -Dsolr.jetty.inetaccess.includes=, -Dsolr.log.dir=/Users/daniilvostrikov/Desktop/solr-8.11.1/server/logs, -Dsolr.log.muteconsole, -Dsolr.solr.home=/Users/daniilvostrikov/Desktop/solr-8.11.1/server/solr, -Duser.timezone=UTC, -XX:+AlwaysPreTouch, -XX:+ExplicitGCInvokesConcurrent, -XX:+ParallelRefProcEnabled, -XX:+PerfDisableSharedMem), **System** (2.59, Physical Memory 98.8%, Swap Space 63.5%, File Descriptor Count 1.7%), **JVM-Memory** (18.2%, 93.09 MB / 512.00 MB), and **Security** (Authentication Plugin, Authorization Plugin, Current Username, User Roles).

Язык запросов в Apache Solr

- Solr Core – это работающий экземпляр индекса Lucene, который содержит все файлы конфигурации Solr, необходимые для его использования. Нам нужно создать Solr Core для выполнения таких операций, как индексация и анализ.
- Приложение Solr может содержать одно или несколько ядер. При необходимости два ядра в приложении Solr могут связываться друг с другом.
- Одним из способов создания ядра является создание ядра без схемы с помощью команды **create**

```
[(base) MacBook-Pro-Daniil-2:solr-8.11.1 daniilvostrikov$ bin/solr create -c solr_sample
WARNING: Using _default configset with data driven schema functionality. NOT RECOMMENDED for production use.
To turn off: bin/solr config -c solr_sample -p 8983 -action set-user-property -property update.autoCreateFields -value false
Created new core 'solr_sample'
```

Язык запросов в Apache Solr

- В Apache Solr можно индексировать (добавлять, удалять, изменять) различные форматы документов, такие как xml, csv, pdf и т. д. Мы можем добавлять данные в индекс Solr несколькими способами: командная строка, веб-интерфейс и клиентский API Java
- Я буду использовать командную строку, а именно пользоваться командой **post** из каталога bin(загрузим файл Mall_customers.csv)

```
[(base) MacBook-Pro-Daniil-2:solr-8.11.1 daniilvostrikov$ bin/post -c solr_sample bin/Mall_Customers.csv
java -classpath /Users/daniilvostrikov/Desktop/solr-8.11.1/dist/solr-core-8.11.1.jar -Dauto=yes -Dc=solr_sample -Ddata=files org.apache.solr.util.SimplePostTool bin/Mall_Customers.csv
SimplePostTool version 5.0.0
Posting files to [base] url http://localhost:8983/solr/solr_sample/update...
Entering auto mode. File endings considered are xml,json,jsonl,csv,pdf,doc,docx,ppt,pptx,xls,xlsx,odt,odp,ods,ott,otp,ots,rtf,htm,html,txt,log
POSTing file Mall_Customers.csv (text/csv) to [base]
1 files indexed.
COMMITting Solr index changes to http://localhost:8983/solr/solr_sample/update...
Time spent: 0:00:01.022
```

Язык запросов в Apache Solr

Перейдем в веб-интерфейс и простым запросом выведем 10 первых записей, чтобы проверить корректность введенной нами команды

The screenshot shows the Apache Solr web interface. On the left, there's a sidebar with various administrative links like Dashboard, Logging, Security, Core Admin, Java Properties, Thread Dump, and others. The main area is titled "Request-Handler (qt)" and contains fields for "q" (set to "*:*"), "q.op" (set to "OR"), and "start, rows" (set to 0, 10). Below these are fields for "fq", "sort", "fl", "df", and "wt". A checkbox for "indent on" is checked. At the bottom, there are checkboxes for "debugQuery", "defType" (set to "lucene"), and "hl", "facet", "spatial", and "spellcheck". A "Raw Query Parameters" field contains "key1=val1&key2=val2". To the right, the browser's address bar shows the URL: `http://localhost:8983/solr/solr_sample/select?indent=true&q.op=OR&q=%3A*`. The page content displays the JSON response from the Solr server, which includes the header, parameters, and a list of 200 documents, each with fields like CustomerID, Genre, Age, id, Annual_Income_k_, Spending_Score_1-100_, and _version_.

```
http://localhost:8983/solr/solr_sample/select?indent=true&q.op=OR&q=%3A*
{
  "responseHeader": {
    "status": 0,
    "QTime": 3,
    "params": {
      "q": "*:*",
      "indent": "true",
      "q.op": "OR",
      "_": "1651578769716"
    }
  },
  "response": {
    "numFound": 200,
    "start": 0,
    "numFoundExact": true,
    "docs": [
      {
        "CustomerID": [1],
        "Genre": ["Male"],
        "Age": [19],
        "id": "205407c1-81db-4ad7-bf76-aaf574dda2e5",
        "Annual_Income_k_": [15],
        "Spending_Score_1-100_": [39],
        "_version_": 1731805702693847040
      },
      {
        "CustomerID": [2],
        "Genre": ["Male"],
        "Age": [21],
        "id": "befa32b6-cab0-48a1-9296-1b3879439f0b",
        "Annual_Income_k_": [15],
        "Spending_Score_1-100_": [81],
        "_version_": 1731805702774587392
      },
      {
        "CustomerID": [3],
        "Genre": ["Female"],
        "Age": [20],
        "id": "3dc674c1-40a5-49eb-9ee9-fce8d4294980",
        "Annual_Income_k_": [16],
        "Spending_Score_1-100_": [6],
        "_version_": 1731805702775635968
      },
      {
        "CustomerID": [4],
        "Genre": ["Female"],
        "Age": [23],
        "id": "c5ff203a-ebac-4015-a4b2-c20f6fid40fb",
        "Annual_Income_k_": [17]
      }
    ]
  }
}
```

Язык запросов в Apache Solr

- **q** - это основной параметр запроса Apache Solr, документы оцениваются по сходству с терминами в этом параметре.
- **fq** - этот параметр представляет запрос фильтра Apache Solr, ограничивая набор результатов документами, соответствующими этому фильтру.
- параметр **start** представляет начальные смещения для результатов страницы, значение этого параметра по умолчанию равно 0.
- **rows** - этот параметр представляет количество документов, которые должны быть получены на странице. Значение по умолчанию для этого параметра – 10.
- **sort** - этот параметр указывает список полей, разделенных запятыми, по которым сортируются результаты запроса.
- **fl** - этот параметр указывает список полей, возвращаемых для каждого документа в наборе результатов.
- **wt** - этот параметр представляет тип ответа, который мы хотели просмотреть в результате(csv, json, xml и т.д.).

Request-Handler (qt)

/select

common

q

q.op

OR

fq

sort

start, rows

0 10

fl

df

wt

The screenshot shows the Apache Solr Request Handler configuration interface. It includes fields for common, q (set to '*.*'), q.op (set to OR), fq (empty), sort (empty), start, rows (0, 10), fl (empty), df (empty), and wt (set to -----). There are also red minus and green plus buttons next to the fq field.

Язык запросов в Apache Solr

- Выведем 10 покупателей женского пола

Request-Handler (qt)

/select

common

q
Genre:Female

q.op
OR

fq

sort

start, rows
0 10

fl

df

wt

indent on

debugQuery

defType
lucene

hl

facet

spatial

spellcheck

Raw Query Parameters
key1=val1&key2=val2

Execute Query

http://localhost:8983/solr/solr_sample/select?indent=true&q.op=OR&q=Genre%3AFemale&rows=10&start=0

```
{
  "responseHeader": {
    "status": 0,
    "QTime": 0,
    "params": {
      "q": "Genre:Female",
      "indent": "true",
      "start": "0",
      "q.op": "OR",
      "rows": "10",
      "_": "1651578769716"
    }
  },
  "response": {
    "numFound": 112,
    "start": 0,
    "numFoundExact": true,
    "docs": [
      {
        "CustomerID": [3],
        "Genre": ["Female"],
        "Age": [20],
        "id": "3dc674c1-40a5-49eb-9ee9-fce8d4294980",
        "Annual_Income_k_": [16],
        "Spending_Score_1-100_": [6],
        "_version_": 1731805702775635968
      },
      {
        "CustomerID": [4],
        "Genre": ["Female"],
        "Age": [23],
        "id": "c5ff203a-ebac-4015-a4b2-c20f6f1d40fb",
        "Annual_Income_k_": [16],
        "Spending_Score_1-100_": [77],
        "_version_": 1731805702776684544
      },
      {
        "CustomerID": [5],
        "Genre": ["Female"],
        "Age": [31],
        "id": "9340ef5d-9cf1-4e0a-8a08-f8bffe769d27",
        "Annual_Income_k_": [17],
        "Spending_Score_1-100_": [40],
        "_version_": 1731805702776684545
      },
      {
        "CustomerID": [6],
        "Genre": ["Female"],
        "Age": [22],
        "id": "02849ff3-85a5-44b4-9a6c-db27fae79125"
      }
    ]
  }
}
```

Язык запросов в Apache Solr

- Узнаем количество покупателей мужчин, которым от 20 до 22 лет(в Response можно увидеть, что их 5)

The screenshot shows the Apache Solr Request Handler interface. On the left, there is a form with various parameters:

- Request-Handler (qt): /select
- common:
 - q: Genre:Male && Age: [20 TO 22]
 - q.op: OR
 - fq: (empty)
 - sort: (empty)
 - start, rows: 0, 10
 - fl: (empty)
 - df: (empty)
 - wt: (empty)
- checkboxes:
 - indent on (checked)
 - debugQuery (unchecked)
 - defType: lucene
 - checkboxes: hl, facet, spatial, spellcheck
- Raw Query Parameters: key1=val1&key2=val2
- Execute Query button

On the right, the response is displayed as a JSON object:

```
http://localhost:8983/solr/solr_sample/select?indent=true&q.op=OR&q=Genre%3AMale%20%26%20Age%3A[20%20TO%2022]&wt=json&start=0&rows=10&indent=true&defType=lucene

{
  "responseHeader": {
    "status": 0,
    "QTime": 0,
    "params": {
      "q": "Genre:Male && Age: [20 TO 22]",
      "indent": "true",
      "start": "0",
      "q.op": "OR",
      "rows": "10",
      "_": "1651578769716"
    },
    "response": {
      "numFound": 5,
      "start": 0,
      "numFoundExact": true,
      "docs": [
        {
          "CustomerID": 2,
          "Genre": ["Male"],
          "Age": [21],
          "id": "befa32b6-cab0-48a1-9296-1b3879439f0b",
          "Annual_Income_k_": [15],
          "Spending_Score_1-100_": [81],
          "_version_": 1731805702774587392
        },
        {
          "CustomerID": 16,
          "Genre": ["Male"],
          "Age": [22],
          "id": "fbff18135-e8be-4972-95de-1e9f731ff341",
          "Annual_Income_k_": [20],
          "Spending_Score_1-100_": [79],
          "_version_": 1731805702782976001
        },
        {
          "CustomerID": 18,
          "Genre": ["Male"],
          "Age": [20],
          "id": "5fedb905-764b-4451-b194-b16ef3f59612",
          "Annual_Income_k_": [21],
          "Spending_Score_1-100_": [66],
          "_version_": 1731805702784024577
        },
        {
          "CustomerID": 100,
          "Genre": ["Male"],
          "Age": [20]
        }
      ]
    }
  }
}
```

Язык запросов в Apache Solr

- Выведем топ-5 самых выгодных для нас покупателей в возрасте 20-30 лет(по Spending Score), причем нам важно знать только их пол и их id

Request-Handler (qt)
/select

— common —

q
Age: [20 TO 30]

q.op
OR

fq

sort
Spending_Score__1-100_desc

start, rows
0 5

fl
Genre, id, Spending_Score__1-100_

df

wt

indent on

debugQuery

defType
lucene

hl

facet

spatial

spellcheck

Raw Query Parameters
key1=val1&key2=val2

Execute Query

http://localhost:8983/solr/solr_sample/select?fl=Genre%2Cid%2CSpending_Score__1-100_

```
{
  "responseHeader": {
    "status": 0,
    "QTime": 1,
    "params": {
      "q": "Age: [20 TO 30]",
      "indent": "true",
      "fl": "Genre, id, Spending_Score__1-100_",
      "start": "0",
      "q.op": "OR",
      "sort": "Spending_Score__1-100_ desc",
      "rows": "5",
      "_": "1651578769716"
    },
    "response": {
      "numFound": 50,
      "start": 0,
      "numFoundExact": true,
      "docs": [
        {
          "Genre": ["Male"],
          "id": "90a069b8-7371-41f1-be64-be13c4c50452",
          "Spending_Score__1-100_": [97]
        },
        {
          "Genre": ["Male"],
          "id": "477bb59d-767e-4cbb-b144-e322383b18dd",
          "Spending_Score__1-100_": [97]
        },
        {
          "Genre": ["Female"],
          "id": "d1501fae-674c-4ab9-8db4-49f7c98ad490",
          "Spending_Score__1-100_": [94]
        },
        {
          "Genre": ["Male"],
          "id": "e6d63b5a-3b1d-45f4-842b-4a9c3840bfce",
          "Spending_Score__1-100_": [92]
        },
        {
          "Genre": ["Female"],
          "id": "42351011-f137-4c8d-a8bd-8c9494d386b7",
          "Spending_Score__1-100_": [89]
        }
      ]
    }
  }
}
```

Язык запросов в Apache Solr

- Также есть удобная функция **facet**. С помощью нее можно, например, вывести количество людей каждого из возрастов, представленных в базе данных(по факту функция **count**). Выведем топ-5 самых часто встречающихся возрастов в нашей базе данных

facet

facet.query

facet.field

Age

facet.prefix

facet.contains

facet.contains.ignoreCase

facet.limit

5

facet.matches

facet.sort

count

facet.mincount

facet.missing

spatial

spellcheck

Raw Query Parameters

key1=val1&key2=val2

```
"_version_":1731805702774587392},  
{  
  "CustomerID": [3],  
  "Genre": [ "Female"],  
  "Age": [20],  
  "id": "3dc674c1-40a5-49eb-9ee9-fce8d4294980",  
  "Annual_Income_k_": [16],  
  "Spending_Score_1-100_": [6],  
  "_version_":1731805702775635968},  
,  
  "CustomerID": [4],  
  "Genre": [ "Female"],  
  "Age": [23],  
  "id": "c5ff203a-ebac-4015-a4b2-c20f6f1d40fb",  
  "Annual_Income_k_": [16],  
  "Spending_Score_1-100_": [77],  
  "_version_":1731805702776684544},  
,  
  "CustomerID": [5],  
  "Genre": [ "Female"],  
  "Age": [31],  
  "id": "9340ef5d-9cf1-4e0a-8a08-f8bffe769d27",  
  "Annual_Income_k_": [17],  
  "Spending_Score_1-100_": [40],  
  "_version_":1731805702776684545}  
},  
"facet_counts":{  
  "facet_queries":{},  
  "facet_fields":{  
    "Age": [  
      "32", 11,  
      "35", 9,  
      "19", 8,  
      "31", 8,  
      "30", 7]},  
    "facet_ranges":{},  
    "facet_intervals":{},  
    "facet_heatmaps":{}}}
```

Язык запросов в Apache Solr

- Выведем топ-5 самых выгодных для нас покупателей в возрасте 20-30 лет(по Spending Score), причем нам важно знать только их пол и их id

Request-Handler (qt)
/select

— common —

q
Age: [20 TO 30]

q.op
OR

fq

sort
Spending_Score__1-100_desc

start, rows
0 5

fl
Genre, id, Spending_Score__1-100_

df

wt

indent on

debugQuery

defType
lucene

hl

facet

spatial

spellcheck

Raw Query Parameters
key1=val1&key2=val2

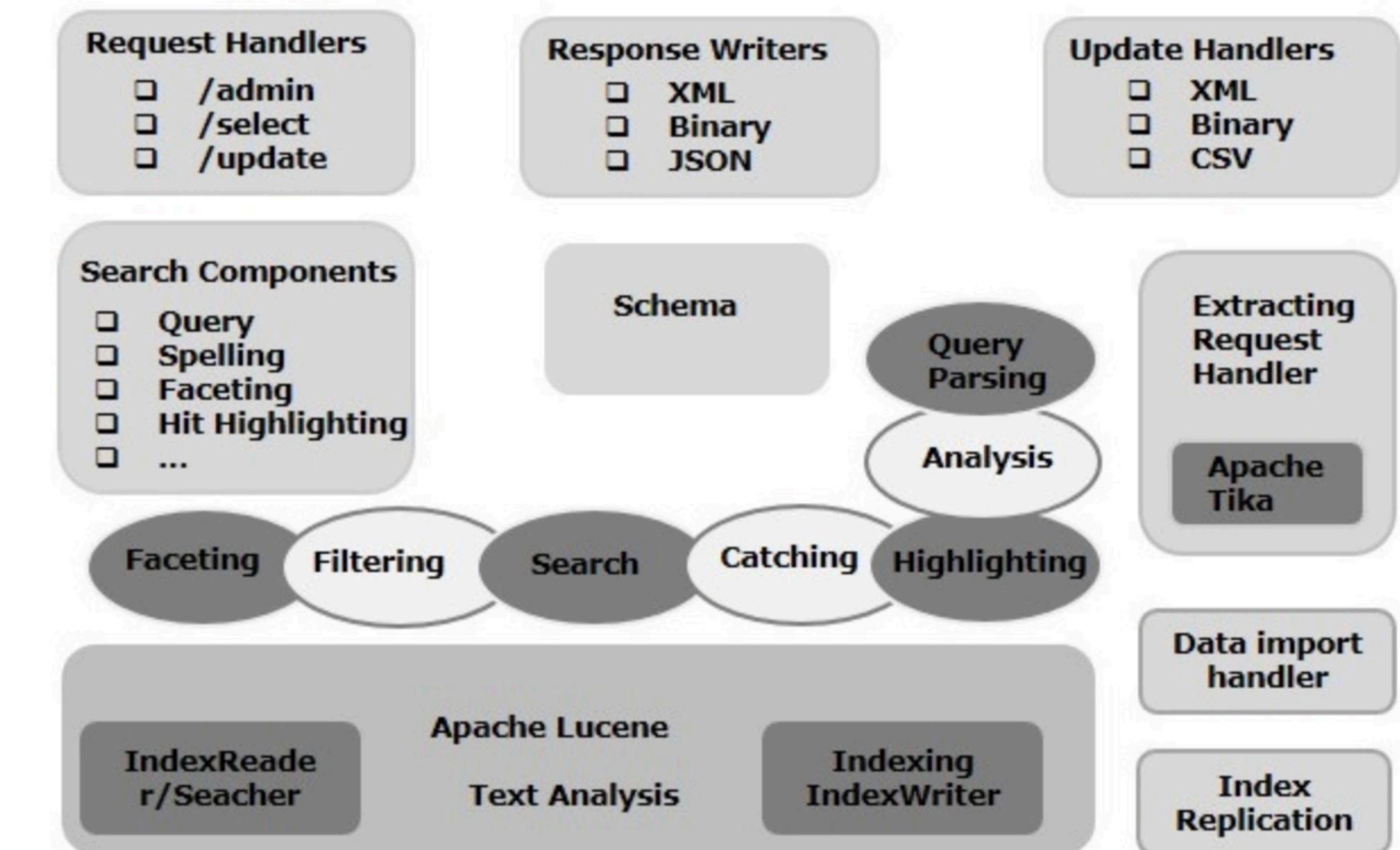
Execute Query

http://localhost:8983/solr/solr_sample/select?fl=Genre%2Cid%2CSpending_Score__1-100_

```
{
  "responseHeader": {
    "status": 0,
    "QTime": 1,
    "params": {
      "q": "Age: [20 TO 30]",
      "indent": "true",
      "fl": "Genre, id, Spending_Score__1-100_",
      "start": "0",
      "q.op": "OR",
      "sort": "Spending_Score__1-100_ desc",
      "rows": "5",
      "_": "1651578769716"
    },
    "response": {
      "numFound": 50,
      "start": 0,
      "numFoundExact": true,
      "docs": [
        {
          "Genre": ["Male"],
          "id": "90a069b8-7371-41f1-be64-be13c4c50452",
          "Spending_Score__1-100_": [97]
        },
        {
          "Genre": ["Male"],
          "id": "477bb59d-767e-4cbb-b144-e322383b18dd",
          "Spending_Score__1-100_": [97]
        },
        {
          "Genre": ["Female"],
          "id": "d1501fae-674c-4ab9-8db4-49f7c98ad490",
          "Spending_Score__1-100_": [94]
        },
        {
          "Genre": ["Male"],
          "id": "e6d63b5a-3b1d-45f4-842b-4a9c3840bfce",
          "Spending_Score__1-100_": [92]
        },
        {
          "Genre": ["Female"],
          "id": "42351011-f137-4c8d-a8bd-8c9494d386b7",
          "Spending_Score__1-100_": [89]
        }
      ]
    }
  }
}
```

Архитектура Apache Solr

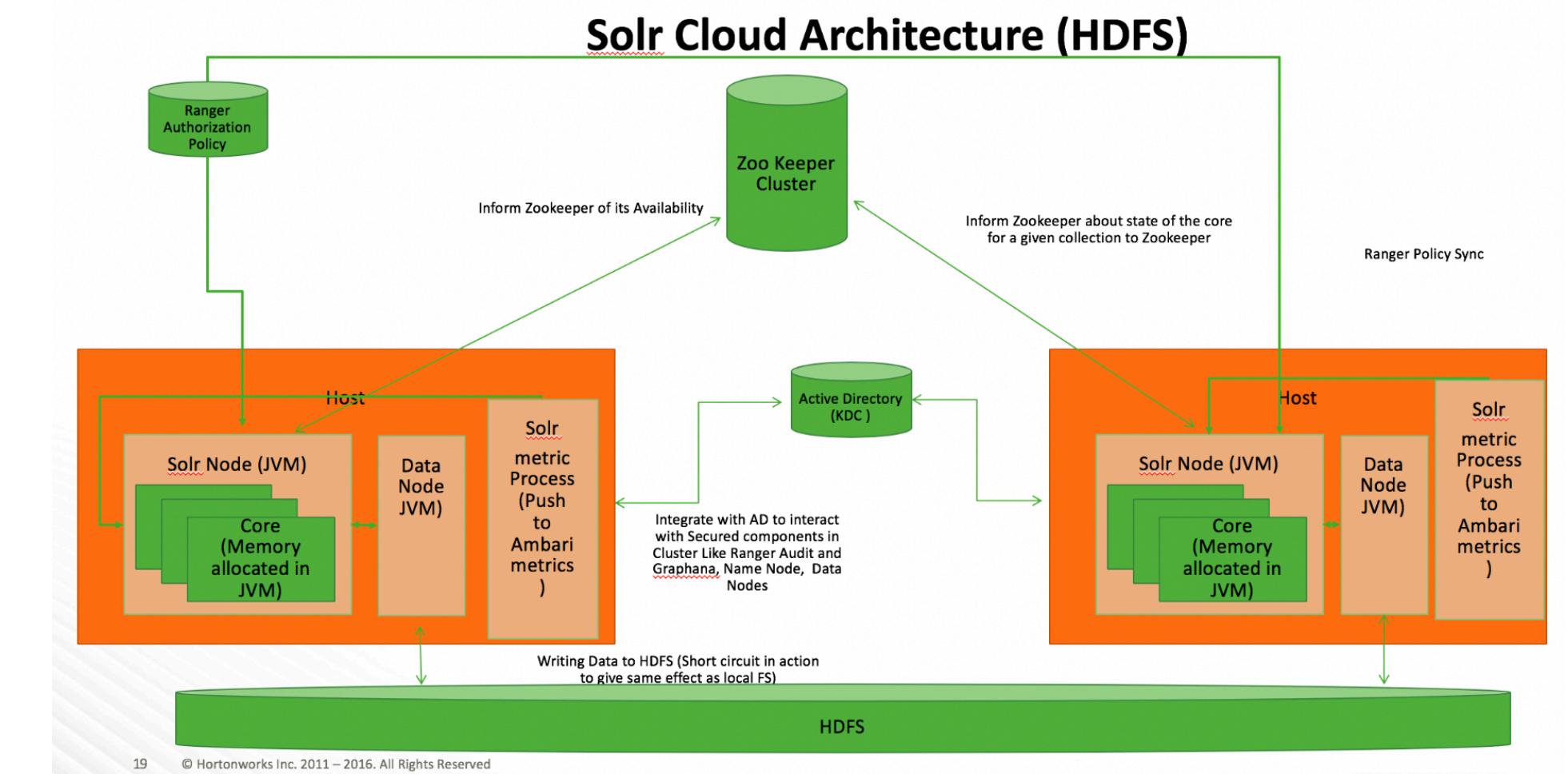
- **Обработчик запросов** - запросы, которые мы отправляем в Apache Solr, обрабатываются этими обработчиками запросов.
- **Компонент поиска** - это тип поиска, предоставляемый в Apache Solr. Это может быть проверка орфографии, запрос, огранка, выделение совпадений и т.д.
- **Парсер запросов** - он анализирует запросы, которые мы передаем Solr, и проверяет запросы на наличие синтаксических ошибок. После анализа запросов он переводит их в формат, понятный Lucene.
- **Средство записи ответов** - это компонент, который генерирует форматированный вывод для пользовательских запросов. Solr поддерживает форматы ответов, такие как XML, JSON, CSV и т.д.
- **Анализатор/токенизатор** - Lucene распознает данные в виде токенов. Apache Solr анализирует содержимое, разделяет его на токены и передает эти токены в Lucene. **Анализатор** в Apache Solr анализирует текст полей и генерирует поток токенов. **Токенизатор** разбивает поток токенов, подготовленный анализатором, на токены.
- **Процессор запросов на обновление** - всякий раз, когда мы отправляем запрос на обновление в Apache Solr, запрос выполняется через набор плагинов (подпись, ведение журнала, индексирование), которые в совокупности известны как **процессор запросов на обновление**.



SolrCloud

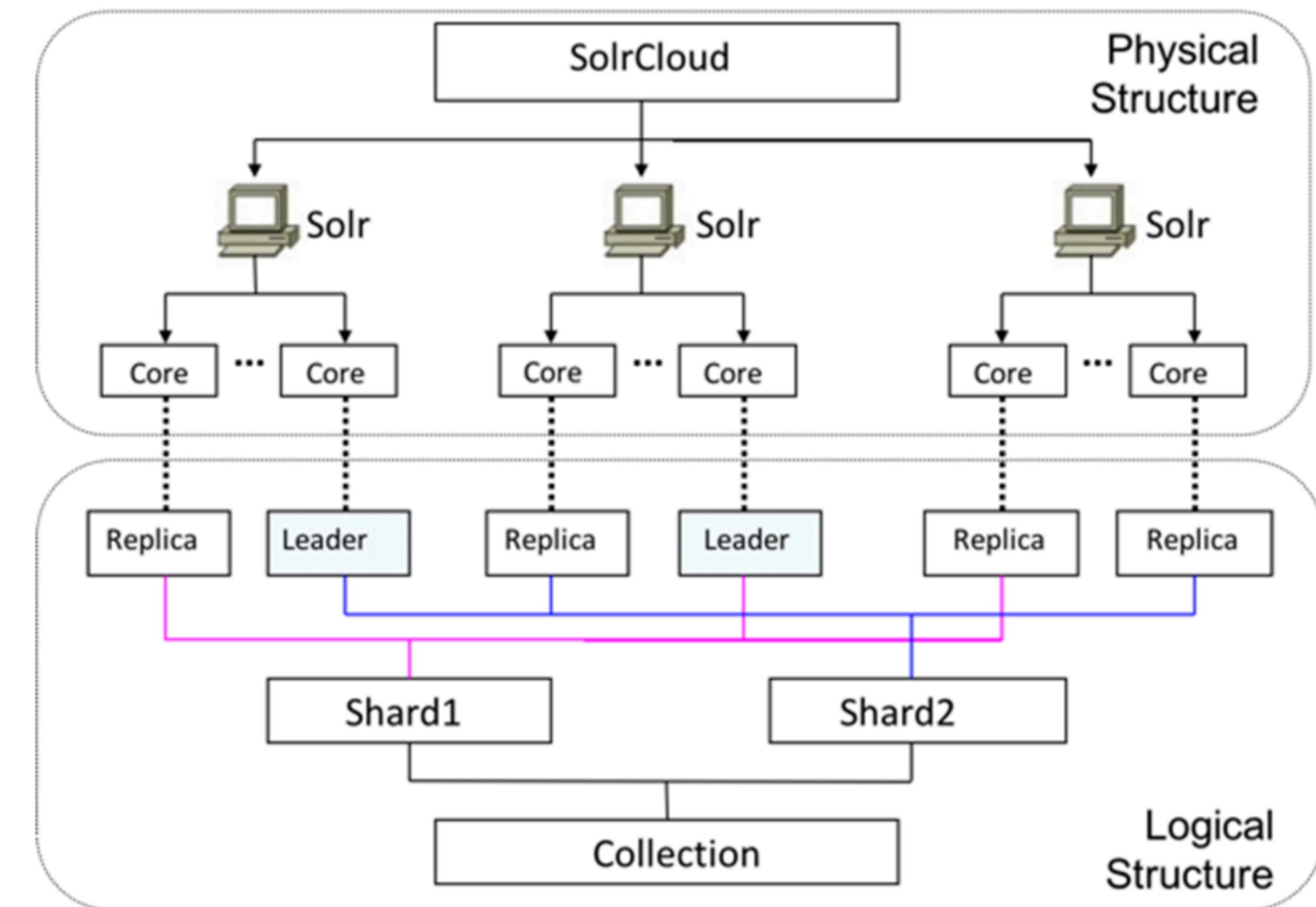
Выше мы обсуждали работу Apache Solr в автономном режиме. Также мы можем установить Solr в распределенном режиме (облачная среда), где Solr устанавливается в шаблоне главный-подчиненный. В распределенном режиме индекс создается на главном сервере и реплицируется на один или несколько подчиненных серверов.

- **Узел** - В облаке Solr каждый отдельный экземпляр Solr рассматривается как **узел**.
- **Кластер** - все узлы среды, объединенные вместе, образуют кластер.
- **Коллекция** - кластер имеет логический индекс, который называется **коллекцией**.
- **Осколок** - это часть коллекции, в которой есть одна или несколько копий индекса.
- **Реплика** - в Solr Core копия осколка, которая работает в узле, называется **репликой**.
- **Лидер** - это также точная копия осколка, которая распределяет запросы облака Solr по оставшимся репликам.
- **ZooKeeper** - это проект Apache, который Solr Cloud использует для централизованной настройки и координации, управления кластером и выбора лидера.



Шардинг с помощью SolrCloud

- SolrCloud обеспечивает шардинг. Существует поддержка автоматического распределения как процесса индексации, так и запросов, а ZooKeeper обеспечивает отказоустойчивость и балансировку нагрузки. Кроме того, каждый сегмент также может иметь несколько реплик для дополнительной надежности.
- В SolrCloud нет ни master'ов, ни slave'ов. Вместо этого каждый осколок состоит по крайней мере из одной физической копии, ровно одна из которых является лидером. Лидеры избираются автоматически, сначала в порядке живой очереди, а затем на основе процесса ZooKeeper, описанного в http://zookeeper.apache.org/doc/trunk/recipes.html#sc_leaderElection.
- Если лидер выходит из строя, одна из других реплик автоматически избирается новым лидером.
- Когда документ отправляется на узел Solr для индексации, система сначала определяет, к какому сегменту принадлежит этот документ, а затем на каком узле в данный момент размещается заголовок для этого сегмента. Затем документ пересыпается текущему лидеру для индексации, а лидер пересыпает обновление всем другим репликам.



Solr и транзакции

- Solr не поддерживает транзакции так, как привыкло большинство разработчиков баз данных(отсутствие изоляции).
- Commit делает все ожидающие изменения всеми клиентами видимыми для новых запросов. Аналогично, откат откатывает все ожидающие изменения всеми клиентами. Не учитывается, какой клиент отправил команду фиксации/отката.
- По этой причине обработка ошибок не должна автоматически приводить к откату. Потому что влияние может быть гораздо шире, чем просто ошибочные данные. И в результате уборка может быть намного сложнее.
- В документации Solr рекомендуется использовать автоматическую фиксацию. Если вы выполняете массовое индексирование, возможно, с несколькими параллельными клиентами, то лучше время от времени автоматически выполнять фиксацию. Это приводит к созданию меньшего количества новых сегментов индекса и в результате к менее фрагментированному индексу в целом.

Методы восстановления в Solr

- Solr предоставляет два подхода к резервному копированию и восстановлению ядер или коллекций Solr, в зависимости от того, как вы используете Solr. Если вы работаете в режиме SolrCloud, вы будете использовать Collections API. Если вы запускаете Solr в автономном режиме, вы будете использовать обработчик репликации.
- Подробнее про эти механизмы написано тут: https://solr.apache.org/guide/6_6/making-and-restoring-backups.html

Solr для анализа данных

- С помощью Solr (+Lucene) также можно запрашивать индексированные данные для анализа и при этом получать ответы невероятно быстро. Применяя Solr подобным образом, вы можете расширить арсенал своего кластера и повысить эффективность использования ресурсов. В некоторой степени Solr может обеспечить возможности, подобные возможностям in-memory NoSQL СУБД (нереляционной СУБД с размещением данных в оперативной памяти).
- Подробнее о том, как проводить анализ и при каких условиях использование Solr для анализа данных и решения бизнес-задач можно прочитать тут: <http://datareview.info/article/how-to-rabota-s-apache-solr/>

Методы защиты в Solr

- Шифрование трафика в/из Solr и между узлами Solr с помощью сертификатов TLS (SSL) предотвращает утечку конфиденциальных данных в сеть.
- Аутентификация, Авторизация и ведение журнала аудита.
- Протоколирование запросов: Solr может дополнительно регистрировать каждый входящий HTTP(s) запрос в стандартном формате NCSA. Вы можете включить ведение журнала запросов, установив `SOLR_REQUESTLOG_ENABLED=true` через переменную среды.
- Контроль доступа по IP: ограничение доступа к сети для определенных хостов, установив `SOLR_IP_WHITELIST/SOLR_IP_BLACKLIST` с помощью переменных среды.
- Включить Диспетчер безопасности: Solr можно запускать в изолированной среде Java Security Manager, установив значение `SOLR_SECURITY_MANAGER_ENABLED=true` через переменную окружения.
- Подробнее о методах защиты можно почитать тут: https://solr.apache.org/guide/8_5/securing-solr.html#encryption-with-tls-ssl-certificates.

Получение доп. информации о Solr

- Изучение языка запросов: https://solr.apache.org/guide/8_11/searching.html
- Документация и обучение: <https://solr.apache.org/resources.html>
- Как быть в курсе происходящего: <https://solr.apache.org/news.html>
- Сообщество, в котором можно быстро получить ответы на интересующие вопросы: <https://solr.apache.org/community.html>