

# Tìm kiếm gần đúng với Apache Solr

## 1. Lời giới thiệu:

### 1.1 Apache solr là gì:

- Giống như người em sinh sau của mình là elastic search, apache solr (gọi tắt là "sôn" ^^) chính là 1 search engine cũng phát triển từ core là lucene.
- Giải quyết bài toán tìm kiếm gần đúng.
- Thích hợp với các loại RDBMS (database quan hệ) hơn là NoSQL, do vậy với data cỡ 20 triệu bản ghi lộn lại sẽ rất thích hợp để dùng solr.

### 1.2 Các công ty dùng solr:

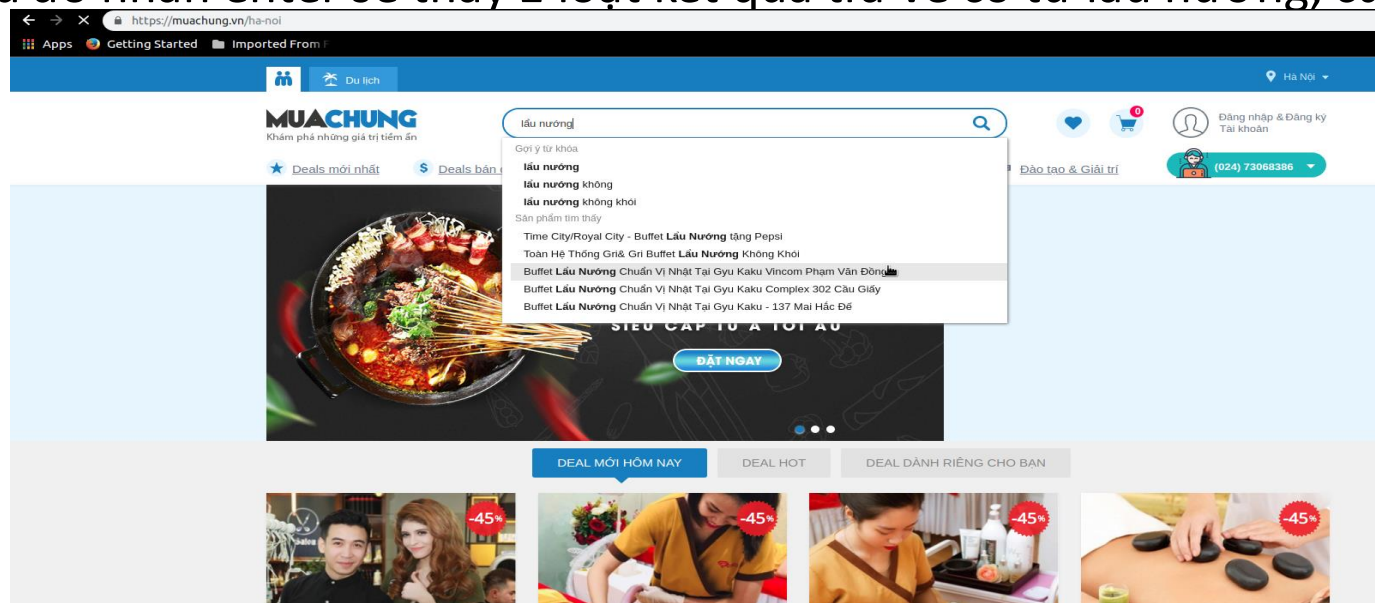
- Ra đời sớm, cổ hơn nên số lượng các công ty dùng solr giảm nhiệt so với elasticsearch, tuy nhiên vẫn có các cái tên lớn vẫn dùng nó: Apple, Cisco, AOL...
- Ở nước ta solr được dùng rất nhiều vì các site cỡ vừa là chủ yếu, do đó khá phù hợp, phải kể đến các site thương mại điện tử của các công ty lớn: Vccorp, adayroi(Vingroup), Sendo...

# Tìm kiếm gần đúng với Apache Solr

## 1. Lời giới thiệu:

### 1.3 Khi nào nên dùng solr:

- Solr sinh ra là để giải quyết 2 bài toán quan trọng: tìm kiếm gần đúng và suggest (gợi ý).
- Giả sử bạn gõ 1 từ khóa ví dụ: "lẩu nướng" trên một site thương mại điện tử (ví dụ muachung.vn) chẳng hạn, bạn thấy phần suggest hiện ra, đó chính là 1 trong 2 chức năng quan trọng của solr, sau đó nhấn enter sẽ thấy 1 loạt kết quả trả về có từ lẩu nướng, cũng như các từ giống với nó:



# Tìm kiếm gần đúng với Apache Solr

## 2. Cài đặt solr:

- Cài solr rất đơn giản, bạn có thể dùng docker hoặc vào trang chủ của nó để download file về.
- Trang chủ là <http://lucene.apache.org/solr/>.
- Chọn file tgz và tải về rồi giải nén ra
- Copy vào 1 thư mục nào đó, chẳng hạn /usr/share. Sau đó vào thư mục **bin** của nó cấp quyền cho file "**solr**" đồng thời chạy bằng câu lệnh sau:

**sudo chmod 777 solr && sudo ./solr start -p 5593 -m 2g**

Trong đó: chmod 777 để cấp quyền, đây là quyền cao nhất.

-p là port, bạn có thể đặt port nào cũng được tùy ý

-m là số memory ram cấp cho solr, thường trong thực tế nên là 4g ->10g, còn ở đây chỉ chạy demo nên cần 2g là đủ

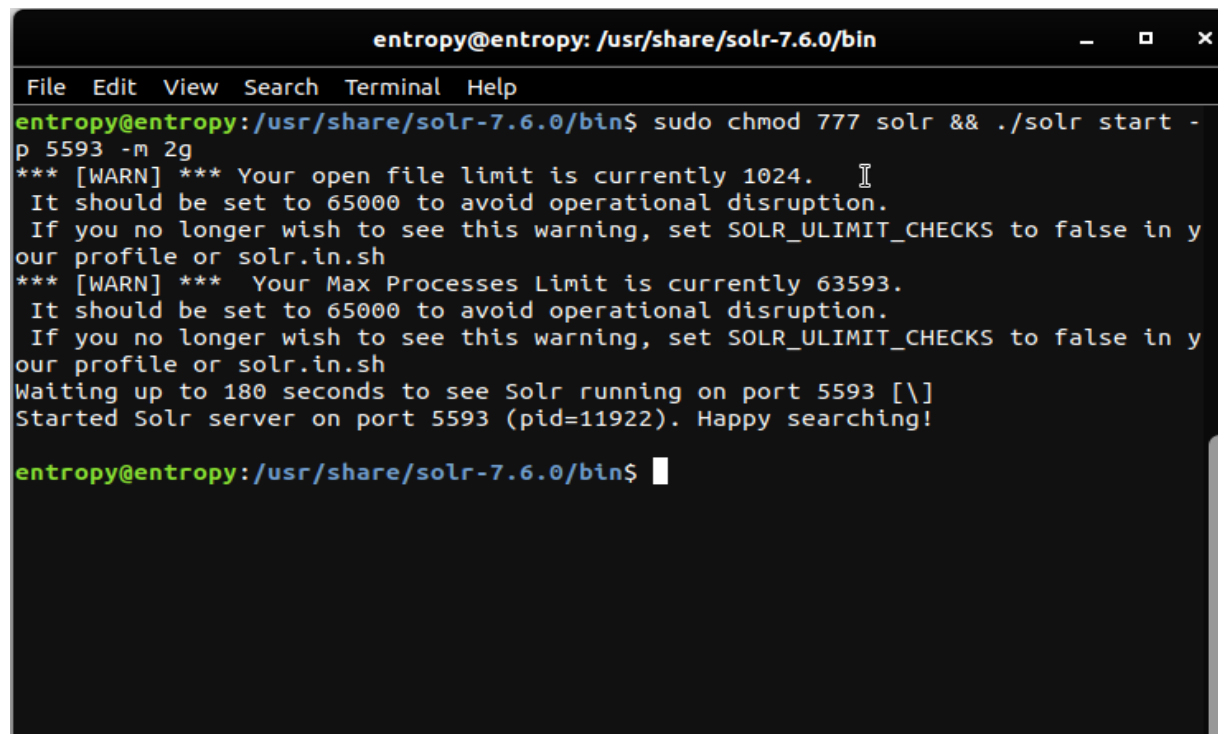
**(Restart bằng lệnh: sudo ./solr restart -p 5593 -m 2g**

**Stop: sudo ./solr stop -p 5593 -m 2g)**

# Tìm kiếm gần đúng với Apache Solr

## 2. Cài đặt solr:

- Chạy thành công sẽ như thế này:(chú ý, khi chạy thực tế trên server cần dùng screen như bên elastic...)

A terminal window titled 'entropy@entropy: /usr/share/solr-7.6.0/bin' with standard window controls. The terminal shows the execution of 'sudo chmod 777 solr && ./solr start -p 5593 -m 2g'. It displays two warning messages about file and process limits, followed by a confirmation that Solr is running on port 5593.

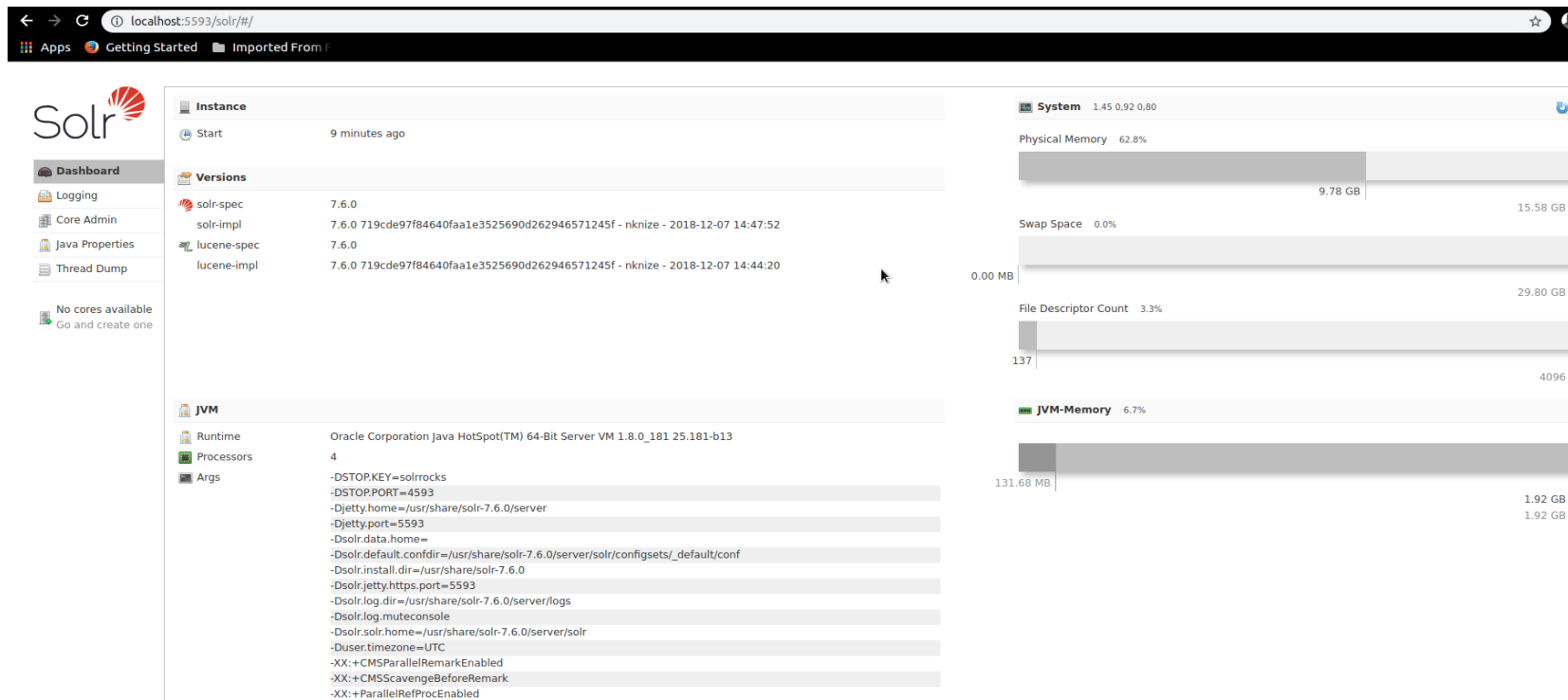
```
entropy@entropy: /usr/share/solr-7.6.0/bin
File Edit View Search Terminal Help
entropy@entropy:/usr/share/solr-7.6.0/bin$ sudo chmod 777 solr && ./solr start -p 5593 -m 2g
*** [WARN] *** Your open file limit is currently 1024.
It should be set to 65000 to avoid operational disruption.
If you no longer 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 63593.
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 5593 [\]
Started Solr server on port 5593 (pid=11922). Happy searching!

entropy@entropy:/usr/share/solr-7.6.0/bin$
```

# Tìm kiếm gần đúng với Apache Solr

## 2. Cài đặt solr:

- Gõ localhost://5593 để xem thành quả:



# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.1 Vấn đề tìm kiếm trên DB:

-> Khá tù túng, cồng kềnh là bạn dùng LIKE query, tuy nhiên nếu bạn muốn search gần đúng thì sao??? Solr sinh ra để làm việc đó.

- Đầu tiên trên solr ta phải tạo ra core là thành phần config để đẩy dữ liệu cũng như query trên solr. Trên solr có thể có n core, mỗi core sẽ là 1 dự án.

- Về lý thuyết là như thế tuy nhiên để đảm bảo performance, không nên chạy quá nhiều core trên 1 solr.

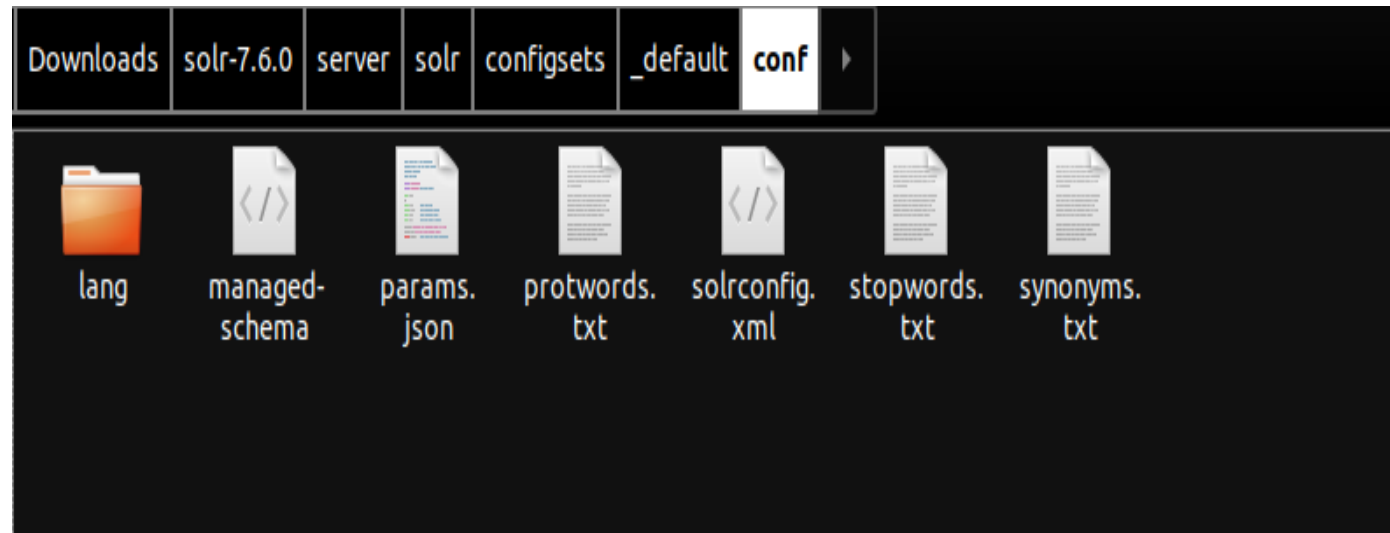
- Lí tưởng nhất là mỗi solr sẽ chạy 1 core.

# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Core là gì:

- Hiểu đơn giản core là thành phần (folder) chứa các file cấu hình để solr có thể lấy data từ DB vào để index cho việc search/suggest.
- Folder solr đã có ví dụ sẵn về cách tạo 1 core, hãy xem trong đường dẫn : [tên solr]/server/solr. Trong thư mục này là 1 vài ví dụ được tích hợp sẵn về cấu trúc của 1 core, ví dụ như hình dưới đây:

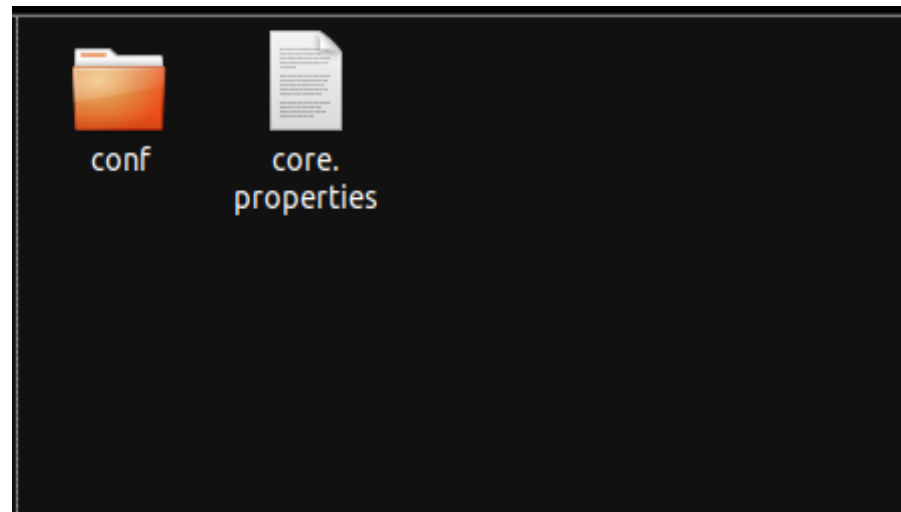


# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Tạo 1 core standard như sau:

- Tạo 1 folder trong đường dẫn sau ([tên solr]/server/solr/) và đặt tên tùy ý, tên này chính là tên core
- Trong folder core này, copy thư mục conf trong example ở slide trước, đây chính là các config cơ bản của 1 core. Ta sẽ copy và dùng, để tránh việc mất thời gian config lại
- Cũng trong folder này, tạo 1 file properties để chứa tên của core: core.properties. nội dung file này chính là "name=tên core"



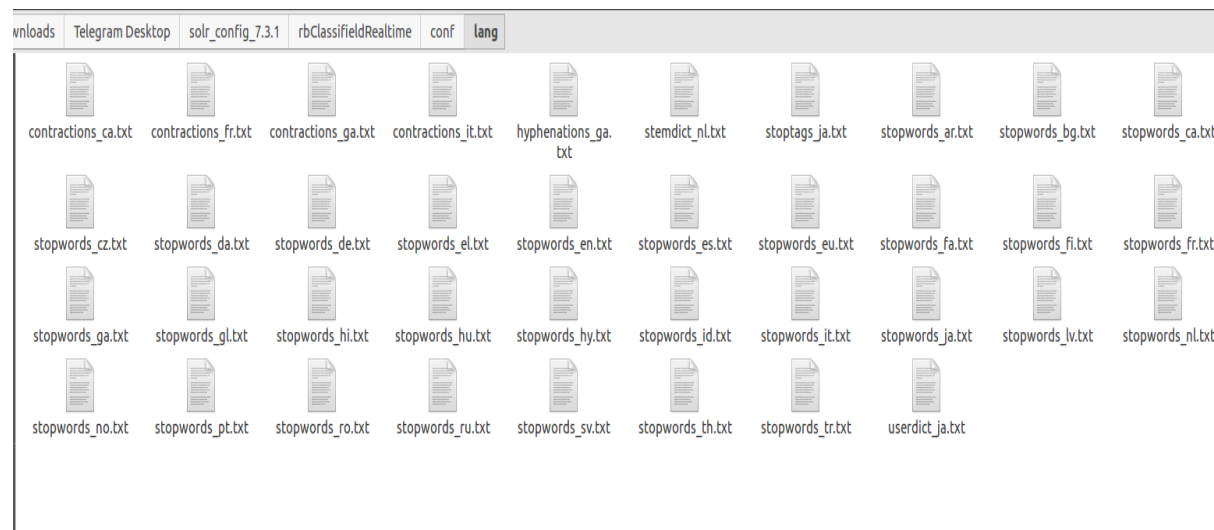


# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Những file cấu hình quan trọng:

- Trong thư mục **conf** , có thư mục **lang** chứa các file text. Các file này chính là các file điều kiện tìm kiếm cấu hình cho solr và có thể customize tùy ý. Ví dụ bạn có thể cấu hình solr bỏ qua việc đánh index các từ tục tĩu trong tiếng việt khi search, bạn sẽ tạo ra 1 file txt chứa các từ tục tĩu đó trong thư mục **lang**. Trong thực tế, chúng ta không quan tâm nhiều lắm tới các file trong **lang**, mà sử dụng mặc định có sẵn. Trừ 1 vài trường hợp như ví dụ trên.

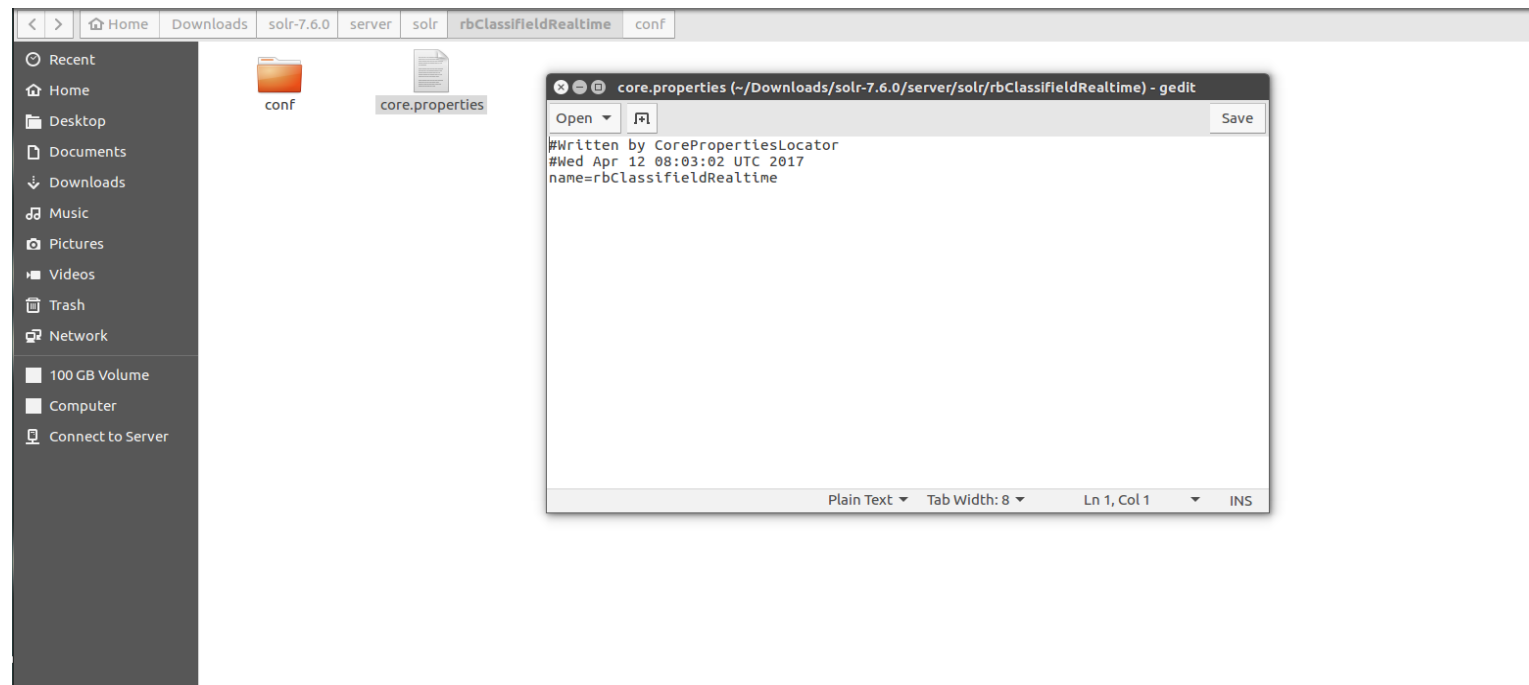


# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Những file cấu hình quan trọng:

- Bây giờ ta xét đến các file đặc biệt quan trọng, để cho đơn giản, tôi đã tạo ra 1 core đặt tên là `rbClassfieldRealtime`, bạn đọc có thể dựa vào đây để tạo core cho riêng mình.



# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Những file cấu hình quan trọng:

3.2.1 File **dataconfig.xml**: Đây là file cấu hình các thông số để đẩy dữ liệu từ DB vào solr.

Có vài tag cần chú ý:

1. query: Khi solr mới chạy lần đầu tiên, tag này sẽ được chạy, đây thực chất là câu query select tất cả các bản ghi khi mà solr chưa có bất kỳ bản ghi nào được index. Và lần đầu tiên đẩy dữ liệu vào solr (lúc này trong solr trống), solr sẽ chạy tag query.

# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Những file cấu hình quan trọng:

```
<?xml version="1.0" ?><dataConfig>
<dataSource batchSize="-1" convertType="true" driver="com.mysql.jdbc.Driver" password="" url="jdbc:mysql:// user="rd" name="database"/>
<dataSource type="URLDataSource" baseUrl="" encoding="UTF-8" connectionTimeout="20000" readTimeout="30000" name="http"/>
<document name="item">
  <entity name="root" pk="ad_id"
    dataSource="database"
    query="SELECT
      ad_id,
      ad_owner,
      ad_cat_id,
      ad_title,
      ad_tags,
      ad_date,
      ad_create_date,
      ad_date_expire,
      ad_date_updated,
      ad_del_date,
      ad_user_del,
      ad_has_picture,
      CAST(ad_is_validated AS SIGNED ) AS ad_is_validated,
      ad_id_pcat,
      ad_id_subcat,
      ad_id_pricecat,
      ad_id_company,
      id_cities,
      id_districts,
      ad_guest_id,
      ad_type,
      ad_choose,
      ad_renew,
      ad_public,
      ad_up_auto,
      ad_vip,
      ad_limitclick,
      ad_ip,
      ad_bad,
      ad_mobile,
      tag_search,
      'meta-content' as meta_content,
      buy_safe,
      is_edit,
      type_check,
      job_cate_id,
      job_model,
      job_contact_address,
      field_extra
    FROM phpclass_ad "
```

# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Những file cấu hình quan trọng:

3.2.1 File **dataconfig.xml**: Đây là file cấu hình các thông số để đẩy dữ liệu từ DB vào solr.

có vài tag cần chú ý:

2. **deltaImportQuery**: Khi có dữ liệu mới, chẳng nhẽ bắt solr phải chạy lại tag query index các bản ghi lại từ đầu. Không, chúng ta chỉ cần index các bản ghi với thời gian lớn hơn hoặc bằng thời gian trước đó trong quá khứ gần nhất. Và tag **deltaImportQuery** sẽ bắt đầu chạy kể từ lần index thứ 2 trở đi.

# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Những file cấu hình quan trọng:

```

<!-- phpclass_ad -->
<entity name="phpclass_ad">
  <deltaImportQuery>=SELECT
    ad_id,
    ad_owner,
    ad_cat_id,
    ad_title,
    ad_tags,
    ad_date,
    ad_create_date,
    ad_date_expire,
    ad_date_updated,
    ad_del_date,
    ad_user_del,
    ad_has_picture,
    CAST(ad_is_validated AS SIGNED ) AS ad_is_validated,
    ad_id_pcat,
    ad_id_subcat,
    ad_id_pricecat,
    ad_id_company,
    id_cities,
    id_districts,
    ad_guest_id,
    ad_type,
    ad_choose,
    ad_renew,
    ad_public,
    ad_up_auto,
    ad_vip,
    ad_limitclick,
    ad_ip,
    ad_bad,
    ad_mobile,
    tag_search,
    'meta-content' as meta_content,
    buy_safe,
    is_edit,
    type_check,
    job_cate_id,
    job_model,
    job_contact_address,
    field_extra
  FROM phpclass_ad where ad_id = '${dih.delta.ad_id}'
  </deltaImportQuery>
  <deltaQuery>=SELECT DISTINCT ad_id from solr_trigger_search where created >= (CONVERT_TZ('${dih.last_index_time}','+00:00','+07:00'))
  </deltaQuery>
  <transformer>=rb.filterarea.Filterarea,rongbay.RongbayExtraFieldTranformer</transformer>
  <field column="data_source" template="jdbc:mysql:/">
</entity>
</document>
</dataConfig>
```

# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Những file cấu hình quan trọng:

3.2.1 File **dataconfig.xml**: Đây là file cấu hình các thông số để đẩy dữ liệu từ DB vào solr.

có vài tag cần chú ý:

3. deltaQuery: Chính là câu query mà solr sẽ thực hiện mỗi khi bạn search:

```
FROM phpclass_ad where ad_id = '${di.h.delta.ad_id}'"
deltaQuery="SELECT DISTINCT ad_id from solr_trigger_search where created >= (CONVERT_TZ('${di.h.last_index_time}','+00:00','+07:00'))"
transformer="rb.filterarea.Filterarea,rongbay.RongbayExtraFieldTransformer">
<field column="data_source" template="jdbc:mysql:/" />

</entity>
</document>
</dataConfig>
```

---

# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Những file cấu hình quan trọng:

3.2.1 File **dataconfig.xml**: Đây là file cấu hình các thông số để đẩy dữ liệu từ DB vào solr.

có vài tag cần chú ý:

4. Ngoài ra bạn có thể filter dữ liệu đầu vào bằng tag "transformer".(VD:validate date, email..)

Với tag này, bạn sẽ phải viết :

1 class extend từ transformer rồi xử lý. Sau đó build thành lib và bỏ vào solr. Lib này sẽ được import trong solrconfig.xml thông qua tag **lib** trỏ tới đường dẫn folder chứa lib đó.

xem ví dụ trong file solrconfig.xml gửi kèm.



# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Những file cấu hình quan trọng:

3.2.2 File **managed-schema.xml**: Tất cả các trường được index hoặc query sẽ được config trong file **managed-schema**. Trong file đó, có thể tùy chỉnh config các trường đó với kiểu phù hợp (int, string, float...) thậm chí là type do mình tự định nghĩa ra. (Thường dùng cho các trường quan trọng để search hoặc suggest). Ngoài ra, thẻ filter sẽ cung cấp cách mà solr sẽ search, hoàn toàn có thể tự config bằng tay. Chi tiết vào 1 file **managed-schema** trong solr gửi kèm, và đọc doc của solr thêm để hiểu thêm cách config.

# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Những file cấu hình quan trọng:

#### 3.2.2 File `solrconfig.xml`

4. Để config search theo source theo ý mình (source mặc định là /query), phải config các thuộc tính để search, kế thừa từ class `solr.StandardRequestHandler`. Để làm được điều này, phải config trong file `solrconfig.xml`. Minh họa như hình dưới đây:

# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Những file cấu hình quan trọng:

#### 3.2.2 File solrconfig.xml

```
<requestHandler class="solr.StandardRequestHandler" name="/rbSearchTag">
<lst name="defaults">
  <str name="qf">ad_title^20</str>
  <str name="bf">recip(ceil(div(div(sub(div(ms(NOW),1000),ad_date),86400),90)),1,1,1)^1000</str>
  <str name="sort">score desc</str>
  <str name="q.alt">*:*</str>
  <str name="defType">dismax</str>
  <str name="echoParams">all</str>
  <str name="mm">4< -1 6< 80%</str>
  <str name="fl">ad_id,ad_owner,ad_cat_id,ad_title,ad_description,ad_tags,ad_date</str>
  <str name="wt">json</str>
  <!-- fq ben php tu truyen len query -->
</lst>
</requestHandler>
```

# Tìm kiếm gần đúng với Apache Solr

## 3. Đẩy data từ DB vào solr để search/suggest:

### 3.2 Những file cấu hình quan trọng:

#### 3.2.2 File `solrconfig.xml`

#### 4. Giải thích một số thuộc tính:

- `qf` (query filter) : config trường cần search. ký tự `^` ám chỉ độ ưu tiên, trường nào `^` cao hơn, trường đó được ưu tiên hơn.
- `bf` (boost filter): nếu không tìm được bản ghi nào thỏa mãn query filter, solr sẽ dùng boost filter để tìm.
- `sort` : sắp xếp (thường là theo điểm `:score`), `desc` : giảm dần, `asc` tăng dần
- `defType`: 2 kiểu search: `dismax`, `edismax`. `Edismax` là mở rộng từ `dismax`, cung cấp nhiều kiểu search hơn. Chi tiết, tìm trên google để hiểu =)))
- `mm`: cái này rất quan trọng, dùng để đặt mức % search đúng. Để hiểu hơn tìm trên google. Như trên hình: nếu text nhập vào dưới 4 ký tự, yêu cầu phải chính xác 100 % khớp với bản ghi mới được return (-1 tương đương với chính xác hoàn toàn, nếu là câu lệnh AND dùng -1, nếu OR dùng 100%). Nếu text nhập vào 6 ký tự trở lên, chỉ cần đúng 80% là return
- `fl`: field list: config các trường cần show ra, cần show ra trường nào thì nhét vào.

# Tìm kiếm gần đúng với Apache Solr

## 4. Chạy index để đẩy dữ liệu:

- Vào giao diện solr từ đường dẫn : localhost://port. ở đây là : localhost://5593. Lúc này sẽ thấy core "rdClassifiedRealtime" hiện ra trong tab "core admin", nếu chưa có tức là config chưa đúng bạn cần dò lỗi do solr bắn ra ngoài để kiểm tra sai sót.
- - Có 2 kiểu index, một là full-index. Tức là các bản ghi sẽ lấy trực tiếp từ db và index all. 2 là delta-index. Delta-index là cải tiến của full index. Các bản ghi sẽ được index từ một điểm thời gian được xác định. Điểm thời gian này chính là thời gian kết thúc index gần đây nhất. Do đó delta index sẽ tăng tốc độ của index (vì không phải index lại những bản ghi trước đó). Tùy vào tình huống mà đặt job theo kiểu full index hay delta index. Thường thì full index chỉ dùng để đẩy data vào từ lần đầu tiên sau khi tạo core (hoặc chỉnh sửa gì đó và cần index lại). Các lần sau, thường sẽ dùng delta index.

Minh họa bởi 2 ảnh sau:

# Tìm kiếm gần đúng với Apache Solr

## 4. Chạy index để đẩy dữ liệu:

/dataimport\_master

/dataimport\_SLOWQUERY\_WITH\_POINT

Command

full-import

☐ Verbose

☒ Clean

☒ Commit

☐ Optimize

☐ Debug

Entity

Start, Rows

0

10

Custom Parameters

key1=val1&key2=val2

Execute

Refresh Status

☐ Auto-Refresh Status

/dataimport\_master

/dataimport\_SLOWQUERY\_WITH\_POINT

Command

delta-import

☐ Verbose

☒ Clean

☒ Commit

☐ Optimize

☐ Debug

Entity

Start, Rows

0

10

Custom Parameters

key1=val1&key2=val2

Execute

Refresh Status

☐ Auto-Refresh Status

# Tìm kiếm gần đúng với Apache Solr

## 4. Chạy index để đẩy dữ liệu:

Mặc định người ta để nút clean tức là mỗi lần click index sẽ xóa (clean) bản ghi cũ đi. Như vậy tùy mục đích sử dụng mà mình sẽ để tích hoặc bỏ tích nút clean đó. Chú ý này rất quan trọng, vì nếu tích clean, data sẽ bị xóa và index lại từ đầu, có thể sẽ lâu hơn so với việc bỏ tích clean.

# Tìm kiếm gần đúng với Apache Solr

## 5. Query:

2 loại query quan trọng thường dùng là : search và suggest. Với search,ta dùng câu truy vấn với "select". còn suggest ta dùng câu truy vấn với "terms".

Chi tiết có thể đọc doc của solr, bản mới :

[https://lucene.apache.org/solr/7\\_6\\_0/index.html](https://lucene.apache.org/solr/7_6_0/index.html)

Cám ơn các bạn đã đọc tới dòng cuối cùng :D , có gì thắc mắc xin liên hệ  
skype:**duybac512** mình sẽ giải đáp các thắc mắc về solr cho các bạn.