Elastic

ارائه دهنـــــــدگان

محمـــد خویـــی | علی مرتضوی منش

اســـتـــــــــاد مربوطـــه | دکتر سید مهدی وحیدی‌پور

بهار97

ElasticSearch یک موتور جستجو و تحلیل منعطف، قدرتمند، متن باز، توزیع شده، دسترسی بالا و بلادرنگ می‌باشد که هسته شاخص گذار آن کتابخانه Lucene می باشد.

الاستیک سرچ یک موتور جستجوی متن با امکانات فراوان از قبیل راحتی استفاده، مقیاس پذیری بالا، فیلترهای متنوع جستجو، تحلیلگرهای اختصاصی و سفارشی برای متون فارسی قبل از ایندکس و ذخیره در دیتابیس، دسته بندی نتایج و امکان گروه بندی و فیلتر ثانویه روی آنها، جستجوی فازی و تقریبی و مانند آنها را دارد.

از ابتدا به منظور استفاده در محیط های توزیع شده پیاده سازی شده است ،جایی که اتکاپذیری و مقیاس پذیری باید وجود داشته باشد، ES توانایی حرکت آسان ماوراء جستجوی کاملا متنی ساده را به شما می‌دهد و رابطه ی اکثر زبان‌های برنامه نویسی معروف را ارائه می دهد، ES وعده های بی حد و حصر استفاده از فناوری جستجو را ارائه می کند.

قابلیت هایی دیگر

- چندین شاخصی
- مبتنی بر سند
- مدیریت مغایرت
- ماندگاری در هر عملیات
- عدم استفاده از قالب ثابت

یک خوشه می‌تواند میزبان چند شاخصی که جدا از هم و یا به صورت یک گروه می‌باشند، باشد.

قابلیت هایی دیگر

- چندین شاخصی
- مبتنی بر سند
- مدیریت مغایرت
- ماندگاری در هر عملیات
- عدم استفاده از قالب ثابت

ذخیره سازی موجودیت‌های پیچیده دنیای واقعی به صورت اسناد JSON انجام می‌گیرد. تمام فیلدها به صورت پیش فرض شاخص‌گذاری می‌شوند و تمام شاخص‌ها در یک درخواست می‌توانند استفاده شوند.

قابلیت هایی دیگر

چندین شاخصی

مبتنی بر سند

مدیریت مغایرت

ماندگاری در هر عملیات

عدم استفاده از قالب ثابت

یک کنترل کننده نسخه به منظور ممانعت از نابودی
اطلاعات در زمان تغییرات همزمان

قابلیت هایی دیگر

- چندین شاخصی
- مبتنی بر سند
- مدیریت مغایرت
- ماندگاری در هر عملیات
- عدم استفاده از قالب ثابت

برای ES در ابتدا امنیت داده مهم است. تغییرات اسناد در گزارش های تراکنش‌ها در چندین نود در خوشه مورد نظر ثبت می‌شود تا امکان از دست رفتن داده به حداقل برسد.

رابطRESTful : استفاده از رابط RESTful به گونه‌ای که از JSON بر روی پروتکل HTTP استفاده می‌کند.

قابلیت هایی دیگر

- چندین شاخصی
- مبتنی بر سند
- مدیریت مغایرت
- ماندگاری در هر عملیات
- عدم استفاده از قالب ثابت

ES به صورت اتوماتیک ساختار داده مورد نظر را از اسناد JSON استخراج کرده و پس از شاخص‌گذاری آن را قابل جستجو قرار می دهد.

سپس بعدها با اعمال دانش خاص منظوره داده های شما، نحوه شاخص‌گذاری داده‌های شما را تعیین می‌کند.

Lucene این پروژه در سال ۱۹۹۹ نوشته شدو در سایت source forge به صورت متن باز ارائه شد و بعدها به بنیاد آپاچی ملحق شد . این نرم افزار یک کتابخانه نرم افزاری بازیابی اطلاعات می باشد . این نرم افزار توسط java نوشته شده است . این نرم افزار به زبان های برنامه نویسی دیگر از جمله دلفی، پرل، C #، C + +، پایتون، روبی و PHP پورت شده است. این نرم افزار برای هر نرم افزاری که نیاز به شاخص گذاری و جستجوی متن به صورت کامل داشته باشد بسیار مفید خواهد بود . Lucene به صورت گسترده ای مورد قبول واقع شده و در موتور جستجوهای اینترنتی و محلی و جستجوی تک سایت مورد استفاده قرار می گیرد.

متن در معماری منطقی هسته Lucene ایده ای نهفته است . این ایده متذکر می شود که یک سند حاوی فیلدهایی از متن هستند . این انعطاف پذیری به API های این نرم افزار اجازه می دهد که مستقل از فرمت فایل باشند . متن از هر فرمت Open Document،HTML،WORD و ... می تواند استخراج شده و شاخص گذاری شود . این عمل بر روی تصاویر غیر قابل ممکن است .

Lucene شامل توابع و کتابخانه هایی برای خزش و پارس نمودن فایل های Html نمی باشد ولی پروژه های دیگری این قابلیت ها را به ارائه نموده اند. اگر به لیست بانکهای اطلاعاتی برتر دنیا هم در سایت db-engines نگاهی بیندازید، این بانک را جزء ده بانک اطلاعاتی مطرح امروزین خواهید یافت.

آشنایی با مفاهیم الاستیک سرچ

همانطور که گفتیم الاستیک سرچ یک موتور منبع باز ، جستجوی کامل متن و تجزیه و تحلیل بسیار مقیاس پذیر است. و به شما امکان ذخیره، جستجو و آنالیز حجم عظیمی از داده‌ها را در زمان اندکی می‌دهد. در ادامه چند اصطلاح پرکاربرد در ES را معرفی و شرح می‌دهیم که برای شروع کار با این موتور الزامی است :

elastic

آشنایی با مفاهیم
الاستیک سرچ

Near Real Time

ES یک پلتفرم جستجوی نزدیک به زمان واقعی است این بدان معنی است که مدت زمانی که طول می‌کشد که یک سند از مرحله ایندکس به مرحله جستجو برسد بسیار ناچیز (Refresh Time) است.

آشنایی با مفاهیم
الاستیک سرچ

Cluster

یک کلاستر مجموعه‌ای از یک یا تعداد بیشتری نود (Server) می‌باشد که این نودها در کنار یکدیگر کل داده ها را نگه داری می‌کنند و امکان جستجو و ایندکس را روی تمام نودها فراهم می‌کنند. یک کلاستر توسط نام خود شناسایی می‌شود و در حالت پیش فرض در ES نام کلاستر ElasticSearch می‌باشد. این نام بسیار مهم است زیرا یک نود تنها با داشتن نام کلاستر مورد نظر میتواند به آن متصل شود و روشن است که روی یک بستر شبکه، نام کلاسترها باید متمایز از دیگری باشد.

آشنایی با مفاهیم
الاستیک سرچ

Node

هر نود می‌تواند یک سیستم جدا باشد که قسمتی از یک کلاستر را تشکیل می‌دهد. نودها محل ذخیره داده‌ها هستند و عملیات Search و Index بروی آنها انجام می‌شود. نودها هم مانند کلاستر توسط نام خود شناسایی می‌شوند و در حالت پیش فرض اگر نام نود توسط کاربر تعیین نشود ES از یک نام به صورت تصادفی از بین اسامی افراد و شخصیت‌های مشهور به آن نسبت می‌دهد و نام هر نود در شناسایی آن ضروری است و در حالت پیشفرض تمام نودهای ایجاد شده به کلاستر پیش فرض ES یعنی ElasticSearch اضافه می‌شوند.

elastic

آشنایی با مفاهیم
الاستیک سرچ

Index

هر ایندکس، مجموعه‌ای از اسناد است که دارای ویژگی‌های تقریبا مشابه هستند و می‌توانیم روی یک کلاستر تنها به تعداد دلخواه ایندکس داشته باشیم و هر ایندکس توسط نام خود شناسایی و تعیین می‌شود و برای انجام عملیات ایندکس، جستجو، حذف وبروزرسانی دانستن نام ایندکس ضروری است.

آشنایی با مفاهیم
الاستیک سرچ

Document

یک سند واحد پایه اطلاعات است که می‌تواند ایندکس شود و اسناد در ES قابل تبدیل شدن به فرمت JSON هستند.

elastic

آشنایی با مفاهیم
الاستیک سرچ

Shard & Replicas

یک ایندکس می‌تواند حجم عظیمی از داده را ذخیره کند که از محدودیت‌های سخت افزاری یک سرور تجاوز می‌کند. بطور مثال یک ایندکس شامل یک ملیارد سند، به فضایی در حدود 1TB جهت ذخیره سازی نیاز دارد که ممکن است بیشتر از ظرفیت یک سرور باشد و یا اگر هم امکان ذخیره سازی این حجم از داده را داشته باشد مدت زمان پاسخ به درخواست‌های جستجو را به مراتب افزایش و در نتیجه باعث کاهش کارایی کلاستر می‌شود. لذا جهت رفع این مشکل ES توانایی تقسیم ایندکس‌ها به چندین قسمت را دارد که هر قسمت یا تکه Shard نامیده می‌شود و در زمان تعریف ایندکس براحتی می‌توان تعداد shard های مورد نیاز را تعیین کرد و هر Shard کاملا کاربردی و مستقل عمل می‌کند و می‌تواند روی هر نود کلاستر قرار بگیرد.

آشنایی با مفاهیم
الاستیک سرچ

Sharding به دو دلیل اصلی مهم است:

حجم داده‌های روی Shardها کم می‌شود و داده‌ها بین Shard ها تقسیم می‌شوند.

امکان اجرای عملیات بصورت توزیع شده روی تمامی نودها فراهم می‌شود که در نتیجه باعث افزایش عملکرد/توان می‌شود.

آشنایی با مفاهیم
الاستیک سرچ

دلایل مهم استفاده از Replication

در محیط‌هایی مانند بستر شبکه که در آن امکان شکست و Fail شدن سیستم در هر زمان را می‌توان انتظار داشت بسیار توصیه می‌شود که یک مکانیزم Fialover جهت حفظ داده‌ها و کارایی سیستم وجود داشته باشد. به همین دلیل ES به شما اجازه می‌دهد تا یک یا چند کپی از هر Shard را روی یک سیستم دیگر به نام Replica Shard داشته باشیم.

دسترس پذیری روی Shard های Fail شده

آشنایی با مفاهیم الاستیک سرچ

درخلاصه این بخش:

هر ایندکس می‌تواند داده‌های خود را روی چند Shard ذخیره کند. یک ایندکس می‌تواند چند و یا صفر Shard کپی داشته باشد .تعداد Shard ها و Replica ها می‌تواند در لحظه تعریف ایندکس تعیین شود و بعد از ایجاد ایندکس، فقط تعداد Replica ها قابل تغییر است. در ES و بصورت پیش فرض هر ایندکس روی ۵ Shard اصلی ایجاد می‌شود و هر Shard اصلی یک Replica دارد.

به طور مثال اگر کلاستر مورد نظر ما ۲ نود داشته باشد، ایندکس شما شامل ۵ Shard اصلی و ۵ Shard Replica است.

انواع رابط (API)های rest full

رابط شاخص

رابط سند

رابط جستجو

رابط کلاستر

رابط (API)
شاخص

ایجاد یک شاخص به همراه مشخص کردن ساختار داده ای سند های آن و تنظیمات مربوطه آن شاخص

مثال

(ایجاد یک‌شاخص)

```
PUT /my_index_name
{
    "settings": {
        "number_of_replicas": 1,
        "number_of_shards": 3,
        "analysis": {},
        "refresh_interval": "1s"
    },
    "mappings": {
        "my_type_name": {
            "properties": {
                "title": {
                "type": "text",
                "analyzer": "english"
                }
            }
        }
    }
}
```

رابط (API)
شاخص

تغییر یک شاخص، بخش تنظیمات

مثال تغییر یک

شاخص

(بخش تنظیمات)

```
PUT /my_index_name/_settings
{
    "index": {
    "refresh_interval": "-1",
    "number_of_replicas": 0
    }
}
```

رابط (API)

شاخص

تغییر یک شاخص، بخش ساختار داده ای

مثال تغییر یک

شاخص

(ساختار داده ای)

```
PUT /my_index_name/_mapping/my_type_name
{
    "my_type_name": {
    "properties": {
        "tag": {
        "type": "keyword"
        }
    }
}
```

رابط (API)

شاخص

دریافت بخش تنظیمات مربوط به یک شاخص:

```
GET /my_index_name/_settings
```

رابط (API)

شاخص

دریافت بخش ساختار داده ای مربوط به یک شاخص:

```
GET /my_index_name/_mapping
```

حذف یک شاخص:

```
DELETE /my_index_name
```

رابط (API)
شاخص

رابط (API)

شاخص

باز کردن و بستن یک شاخص جهت مدیریت مصرف حافظه و پردازنده:

```
1    POST /my_index_name/_close
2    POST /my_index_name/_open
```

رابط (API)

سند

ایجاد یک سند با قابلیت ساخت کلید اصلی خودکار

مثال

(ایجاد یک‌سند)

```
POST /my_index_name/my_type_name
{
"title":
    "Elastic is funny",
    "tag": [    "lucene"  ]
}
```

رابط (API) سند

ایجاد و تغییر یک سند مشخص بر
اساس یک کلید مشخص :

مثال

(ایجاد و تغییر یک‌سند)

```
PUT /my_index_name/my_type_name/12abc
{
    "title":
        "Elastic is funny",
    "tag": [    "lucene"  ]
}
```

رابط (API) سند

حذف یک سند مشخص بر اساس یک کلید مشخص:

```
DELETE /my_index_name/my_type_name/12abc
```

رابط (API)
جستجو

جستجو بر روی تمامی شاخص ها و
تمامی انواع سند موجود در آنها:

```
GET /_search
{
    query{}
}
```

رابط (API)

جستجو

جستجو برروی یک شاخص خاص با مشخص کردن نوع اسناد و یا مشخص نکردن نوع اسناد(جستو جو بر روی تمام انواع موجود سند در شاخص):

مثال

```
GET /my_index_name/_search
{
    query{}

}
GET /my_index_name/type_name_1,type_name_2
{
    query{}

}
```

نمونه از رابط search

رابط (API)

جستجو

```
GET /_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "title": "smith"
          }
        }
      ],
      "filter": [
        {
          "exists": {
            "field": "title"
          }
        }
      ]
    }
  },
   "size": 20,
  "from": 100,
  "_source": [
    "title",
    "id"
  ],
  "sort": [
    {
      "_id": {
        "order": "desc"
      }
    }
  ]
}
```

elastic

فیلم‌های آموزشی

جهت دریافت فیلم های آموزشی به کانال تلگرامی

@elastic_learn

مراجعه کنید.

برخی موارد پیشرفته تر

بخش تجزیه و تحلیل (analysis) در شاخص:

تعریف :

تعریف سایت رهنما:

Analysis is the process of converting text, like the body of any email, into *tokens* or *terms* which are added to the inverted index for searching. Analysis is performed by an *analyzer* which can be either a built-in analyzer or a `custom` analyzer defined per index.

مشاهده ی نحوه ی عملکر یکی از تحلیلگر های از پیش تعریف شده

"The QUICK brown foxes jumped over the lazy dog!"

[quick, brown, fox, jump, over, lazi, dog]

ساختارهای اصلی یک تحلیلگر:

فیلتر گذاری کارکتر ها(character filter)

متن پایه و خام را دریافت میکند و با تغییر ویا حذف ویا اضافه کردن کارکتر ها متن پایه برای فرایند تشخیص کلمات آماده می کند .

تعریف سایت راهنما:

A character filter receives the original text as a stream of characters and can transform the stream by adding, removing, or changing characters. For instance, a character filter could be used to convert Hindu-Arabic numerals (٠١٢٣٤٥٦٧٨٩) into their Arabic-Latin equivalents (0123456789), or to strip HTML elements like <b> from the stream.

An analyzer may have zero or more character filters, which are applied in order.

تشخیص و استخراج کلمات(tokenizer)

کاراکتر های ورودی را دریافت میکند و با تشخیص کلمه، کلمه شناخته شده را جدا سازی می کند

تعریف سایت راهنما:

A tokenizer receives a stream of characters, breaks it up into individual tokens (usually individual words), and outputs a stream of tokens. For instance, a whitespace tokenizer breaks text into tokens whenever it sees any whitespace. It would convert the text "Quick brown fox!" into the terms [Quick, brown, fox!].

The tokenizer is also responsible for recording the order or position of each term and the start and end character offsets of the original word which the term represents.

An analyzer must have exactly one tokenizer.

فیلتر گذاری بر روی کلمات(token filter)

کلمات را دریافت میکند و فیلتر هایی را بر روی آن اعمال میکند که میتواند موجب تغییر حذف و یا ایجاد کلمات شود.

تعریف سایت راهنما:

A token filter receives the token stream and may add, remove, or change tokens. For example, a lowercase token filter converts all tokens to lowercase, a stop token filter removes common words (stop words) like the from the token stream, and a synonym token filter introduces synonyms into the token stream.

Token filters are not allowed to change the position or character offsets of each token.

An analyzer may have zero or more token filters, which are applied in order.

چند مورد از انواع تحلیلگر های از پیش تعریف شده :

Standard Analyzer

The standard analyzer divides text into terms on word boundaries, as defined by the Unicode Text Segmentation algorithm. It removes most punctuation, lowercases terms, and supports removing stop words.

Simple Analyzer

The simple analyzer divides text into terms whenever it encounters a character which is not a letter. It lowercases all terms.

Whitespace Analyzer

The whitespace analyzer divides text into terms whenever it encounters any whitespace character. It does not lowercase terms.

تغییر یک نمونه تحلیل گر از پیش تعریف شده:

```
PUT my_index
{
  "settings": {
    "analysis": {
      "analyzer": {
        "std_english": {
          "type":      "standard",
          "stopwords": "_english_"
        }
      }
    }
  },
  "mappings": {
    "_doc": {
      "properties": {
        "my_text": {
          "type":     "text",
          "analyzer": "standard",
          "fields": {
            "english": {
              "type":     "text",
              "analyzer": "std_english"
            }
          }
        }
      }
    }
  }
}
```

انواع حالت های انتساب یک تحلیلگر:

- انتساب به یک فیلد خاص یک سند
- انتساب به یک سند
- انتساب به یک شاخص

انتساب یک تحلیلگر (analizer) برای یک فیلد خاص
از یک نوع سند خاص در یک شاخص:

```
PUT my_index
{
  "mappings": {
    "_doc": {
      "properties": {
        "title": {
          "type":     "text",
          "analyzer": "standard"
        }
      }
    }
  }
}
```

ایجاد یک نمونه تحلیل گر (analyzer) :

```
PUT /english_example
{
  "settings": {
    "analysis": {
      "filter": {
        "english_stop": {
          "type":       "stop",
          "stopwords":  "_english_"
        },
        "english_stemmer": {
          "type":       "stemmer",
          "language":   "english"
        }
      },
      "analyzer": {
        "my_english": {
          "char_filter":  ["html_strip"],
          "tokenizer":  "standard",
          "filter": [
            "lowercase",
            "english_stop",
            "english_stemmer"
          ]
        }
      }
    }
  }
}
```

مراجـــع

- www.bigdata-ir.com
- www.tutorialspoint.com/elasticsearch/elasticsearch_analysis.htm
- www.tutorialspoint.com/elasticsearch/elasticsearch_document_apis.htm
- www.tutorialspoint.com/elasticsearch/elasticsearch_index_apis.htm
- www.tutorialspoint.com/elasticsearch/elasticsearch_mapping.htm
- www.tutorialspoint.com/elasticsearch/elasticsearch_search_apis.htm

باتشکر از توجه شما