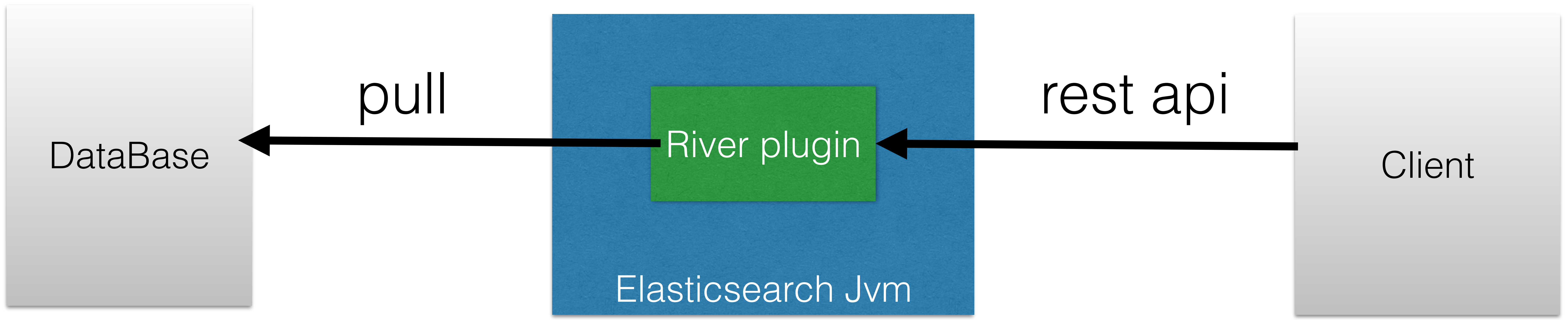
Elasticsearch-jdbc介绍 及基于binlog增量同步方案

卢栋@杭州码耘网络

rivers

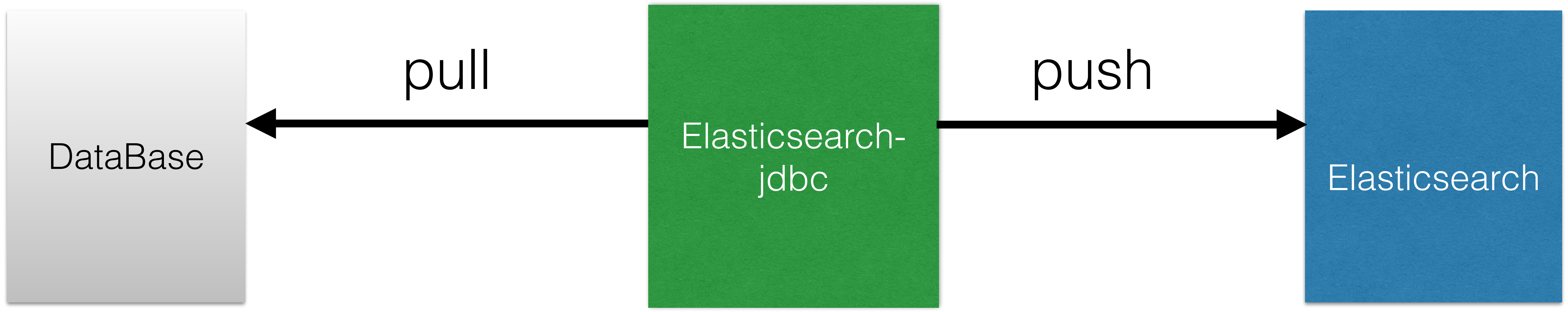


缺点

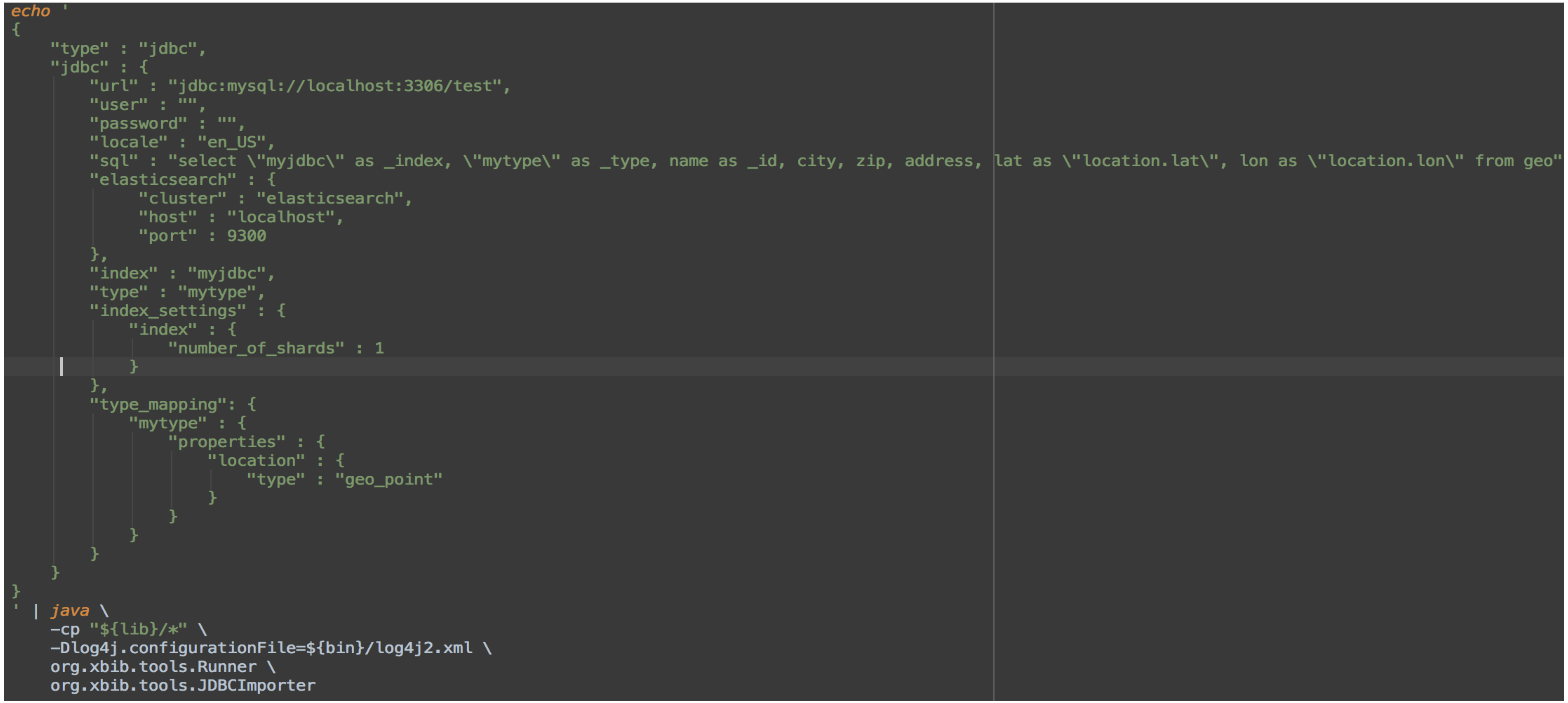
- elastic集群的规模直接决定了同步任务的并行度
- 不能对数据做filter
- 由于和elastic共享一个jvm，会带来额外的内存、网络、io的消耗，影响elastic集群的稳定性

Rivers removed in Elasticsearch 2.0

elasticsearch-jdbc

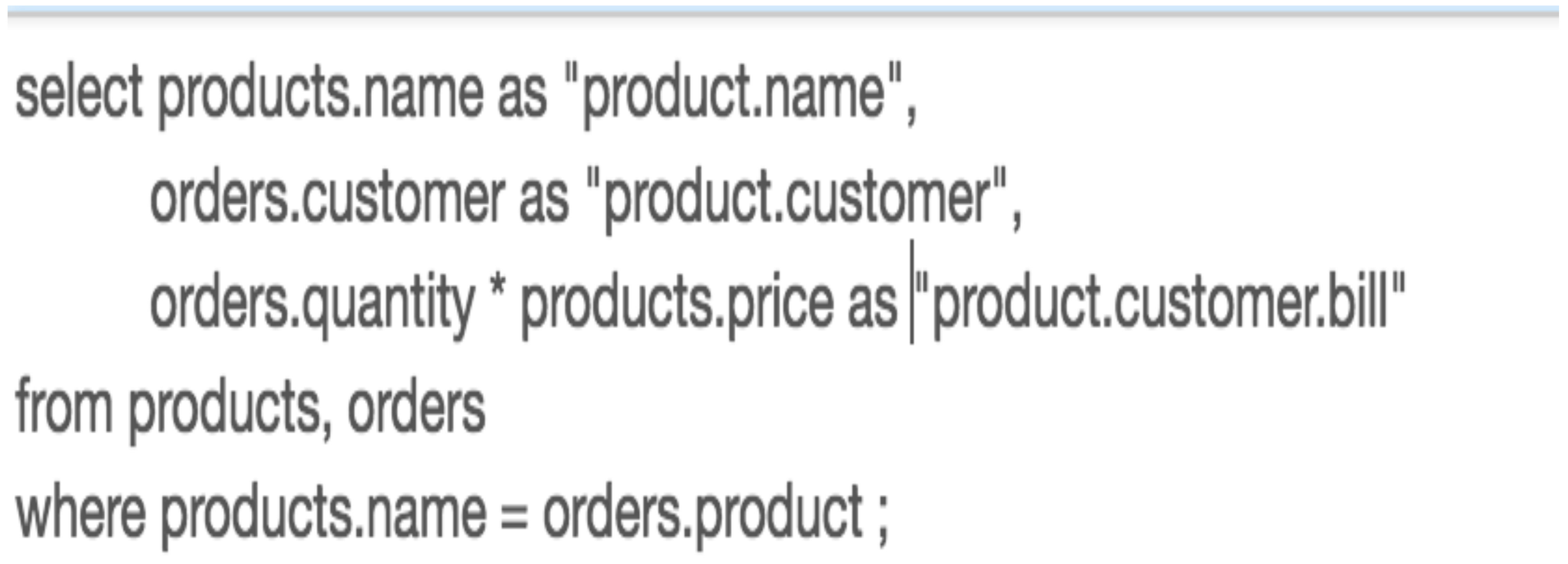


config demo



```
echo '
{
    "type" : "jdbc",
    "jdbc" : {
        "url" : "jdbc:mysql://localhost:3306/test",
        "user" : "",
        "password" : "",
        "locale" : "en_US",
        "sql" : "select \"myjdbc\" as _index, \"mytype\" as _type, name as _id, city, zip, address, lat as \"location.lat\", lon as \"location.lon\" from geo"
        "elasticsearch" : {
             "cluster" : "elasticsearch",
             "host" : "localhost",
             "port" : 9300
        },
        "index" : "myjdbc",
        "type" : "mytype",
        "index_settings" : {
            "index" : {
                "number_of_shards" : 1
            }
        },
        "type_mapping": {
            "mytype" : {
                "properties" : {
                    "location" : {
                        "type" : "geo_point"
                    }
                }
            }
        }
    }
}
' | java \
    -cp "${lib}/*" \
    -Dlog4j.configurationFile=${bin}/log4j2.xml \
    org.xbib.tools.Runner \
    org.xbib.tools.JDBCImporter
```

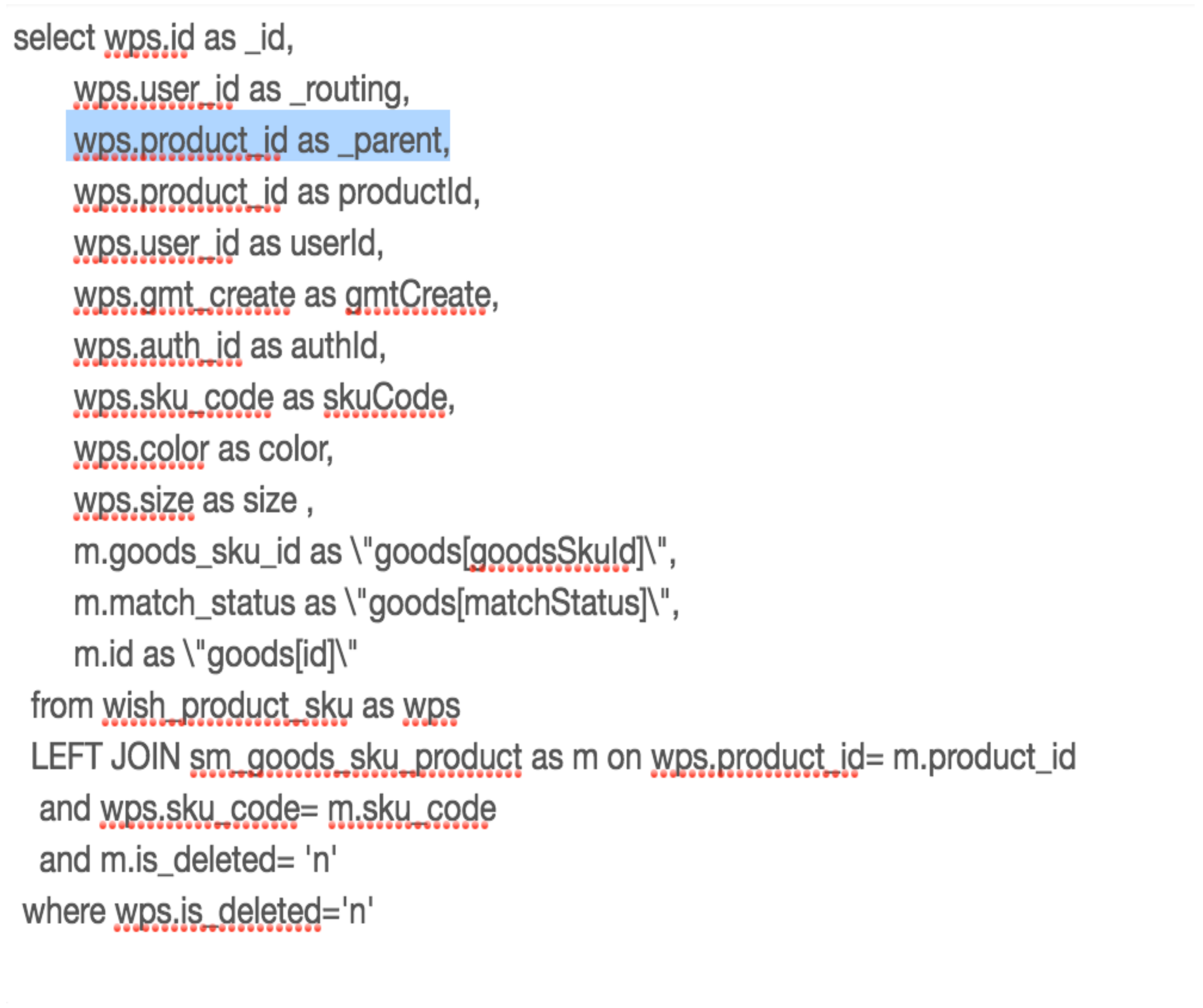
nested object



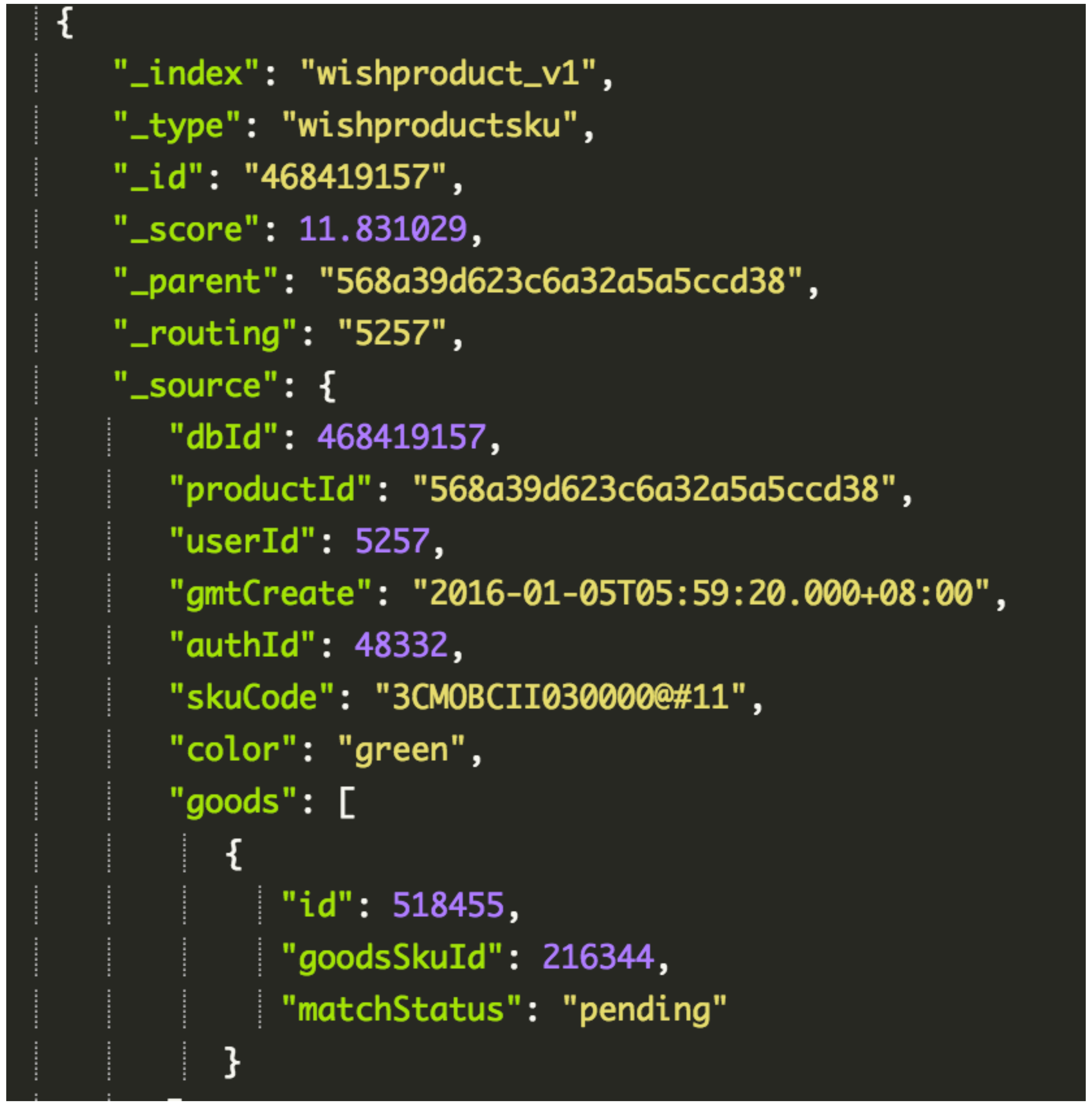
```
select products.name as "product.name",
        orders.customer as "product.customer",
        orders.quantity * products.price as "product.customer.bill"
from products, orders
where products.name = orders.product ;
```

- id=0 {"product":{"name":"Apples","customer":{"bill":1.0,"name":"Big"}}}

parent-child

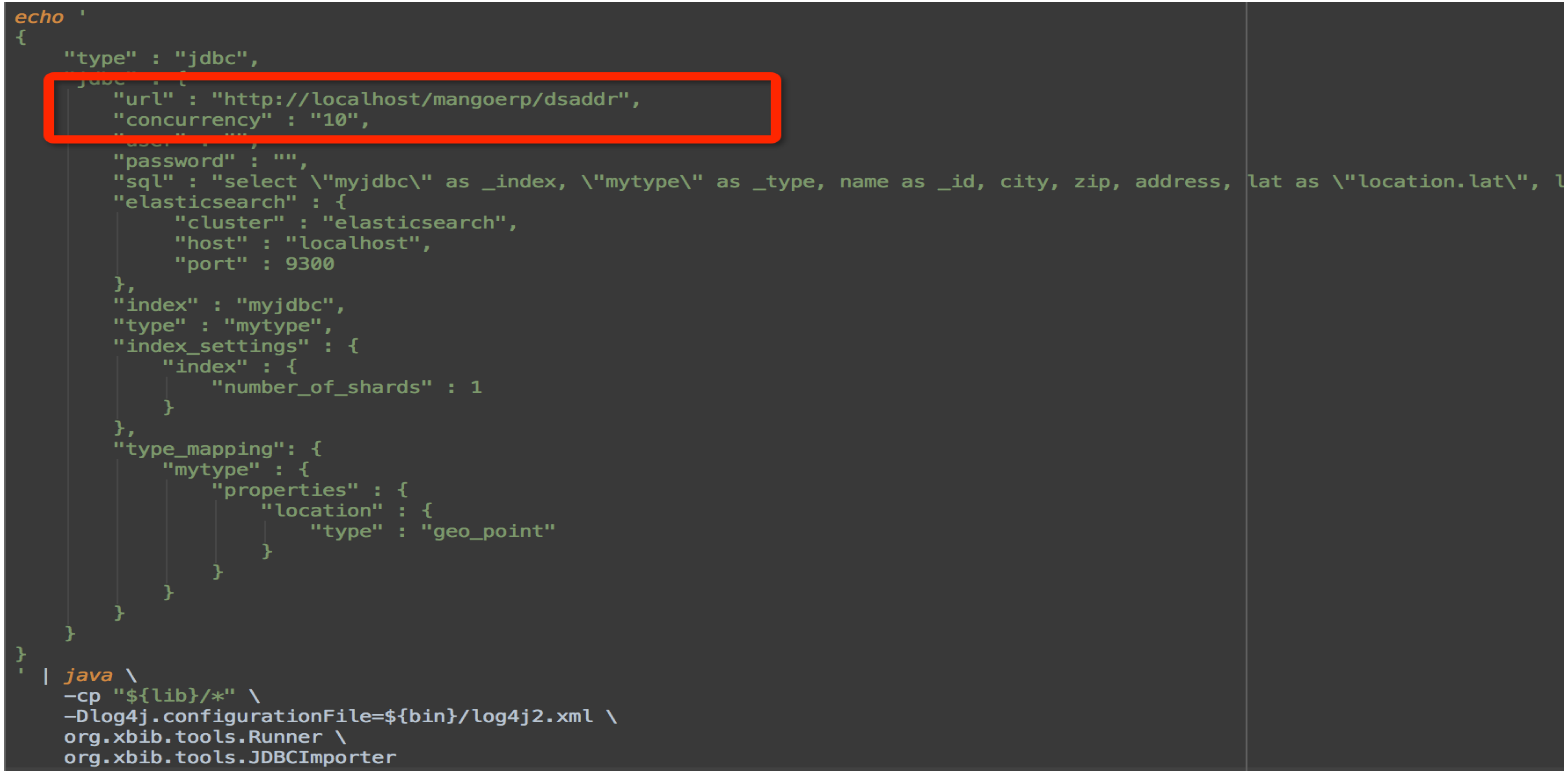


```
select wps.id as _id,
      wps.user_id as _routing,
      wps.product_id as _parent,
      wps.product_id as productId,
      wps.user_id as userId,
      wps.gmt_create as gmtCreate,
      wps.auth_id as authId,
      wps.sku_code as skuCode,
      wps.color as color,
      wps.size as size ,
      m.goods_sku_id as \"goods[goodsSkuId]\",
      m.match_status as \"goods[matchStatus]\",
      m.id as \"goods[id]\"
  from wish_product_sku as wps
  LEFT JOIN sm_goods_sku_product as m on wps.product_id= m.product_id
   and wps.sku_code= m.sku_code
   and m.is_deleted= 'n'
 where wps.is_deleted='n'
```



```
{
    "_index": "wishproduct_v1",
    "_type": "wishproductsku",
    "_id": "468419157",
    "_score": 11.831029,
    "_parent": "568a39d623c6a32a5a5ccd38",
    "_routing": "5257",
    "_source": {
        "dbId": 468419157,
        "productId": "568a39d623c6a32a5a5ccd38",
        "userId": 5257,
        "gmtCreate": "2016-01-05T05:59:20.000+08:00",
        "authId": 48332,
        "skuCode": "3CMOBCII030000@#11",
        "color": "green",
        "goods": [
            {
                "id": 518455,
                "goodsSkuId": 216344,
                "matchStatus": "pending"
            }
```

support muliti jdbc url



```
echo '
{
    "type" : "jdbc",
    "jdbc" : {
        "url" : "http://localhost/mangoerp/dsaddr",
        "concurrency" : "10",
        "user" : "",
        "password" : "",
        "sql" : "select \"myjdbc\" as _index, \"mytype\" as _type, name as _id, city, zip, address, lat as \"location.lat\", l
        "elasticsearch" : {
             "cluster" : "elasticsearch",
             "host" : "localhost",
             "port" : 9300
        },
        "index" : "myjdbc",
        "type" : "mytype",
        "index_settings" : {
            "index" : {
                "number_of_shards" : 1
            }
        },
        "type_mapping": {
            "mytype" : {
                "properties" : {
                    "location" : {
                        "type" : "geo_point"
                    }
                }
            }
        }
    }
}
' | java \
    -cp "${lib}/*" \
    -Dlog4j.configurationFile=${bin}/log4j2.xml \
    org.xbib.tools.Runner \
    org.xbib.tools.JDBCImporter
```

json – object

需求场景:

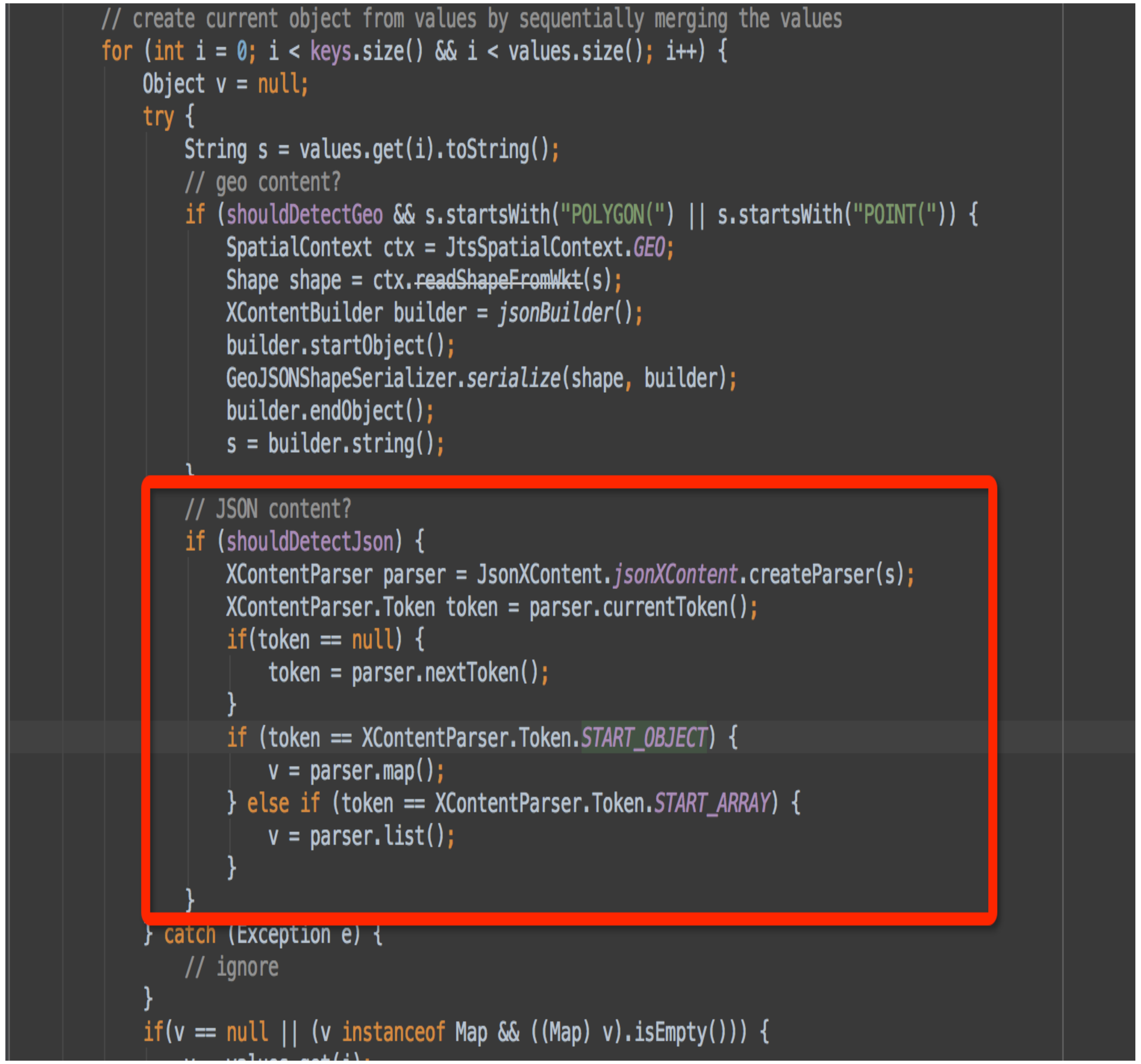
- 数据库中的字段存的是一个json字符串，存储到elasticsearch上也是一个字符串
- 举例：columnKey : sku   columnValue ： [{“name”:”金属颜色","value":"镀白金"}]

现象:

- elasticsearch报了大量IllegalArgumentException[unknown property [name]]] 的异常

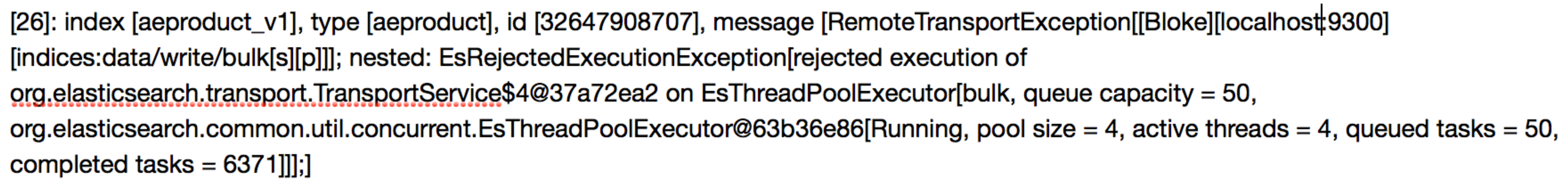
解决方法:

- detect_json - if json structures in SQL columns should be parsed when constructing JSON documents. Default is true



```
// create current object from values by sequentially merging the values
for (int i = 0; i < keys.size() && i < values.size(); i++) {
    Object v = null;
    try {
        String s = values.get(i).toString();
        // geo content?
        if (shouldDetectGeo && s.startsWith("POLYGON(") || s.startsWith("POINT(")) {
            SpatialContext ctx = JtsSpatialContext.GEO;
            Shape shape = ctx.~~readShapeFromWkt~~(s);
            XContentBuilder builder = jsonBuilder();
            builder.startObject();
            GeoJSONShapeSerializer.serialize(shape, builder);
            builder.endObject();
            s = builder.string();
        }
        // JSON content?
        if (shouldDetectJson) {
            XContentParser parser = JsonXContent.jsonXContent.createParser(s);
            XContentParser.Token token = parser.currentToken();
            if(token == null) {
                token = parser.nextToken();
            }
            if (token == XContentParser.Token.START_OBJECT) {
                v = parser.map();
            } else if (token == XContentParser.Token.START_ARRAY) {
                v = parser.list();
            }
        }
    } catch (Exception e) {
        // ignore
    }
    if(v == null || (v instanceof Map && ((Map) v).isEmpty())) {
```

threadpool reject task



```
[26]: index [aeproduct_v1], type [aeproduct], id [32647908707], message [RemoteTransportException[[Bloke][localhost:9300]
[indices:data/write/bulk[s][p]]]; nested: EsRejectedExecutionException[rejected execution of
org.elasticsearch.transport.TransportService$4@37a72ea2 on EsThreadPoolExecutor[bulk, queue capacity = 50,
org.elasticsearch.common.util.concurrent.EsThreadPoolExecutor@63b36e86[Running, pool size = 4, active threads = 4, queued tasks = 50,
completed tasks = 6371]]];]
```

- For bulk operations. Thread pool type is fixed with a size of # of available processors, queue_size of 50.
- 调整bulk thread pool: thread size: 15   queue size:1000
- elasticsearch threadpool document

OutOfMemory

现象:

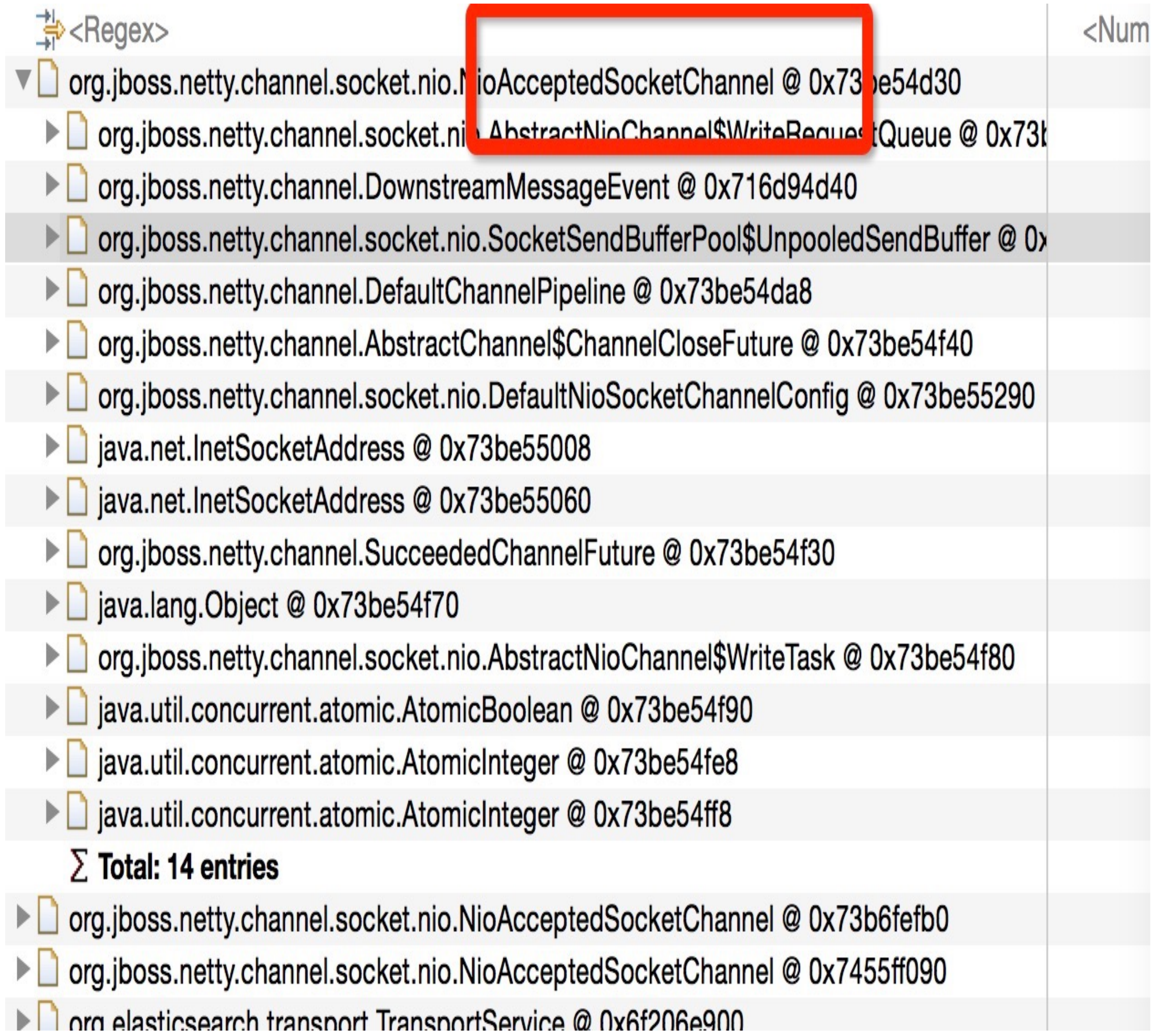
- 迁移过程elasticsearch端出现OutOfMemoryError的错误

配置:

- concurrency: 100
- max_bulk_action: 10000 (default 10000)
- max_concurrrent_bulk_requests: 8 (2 * number of cpu cores)
- ignoreBulkErrors: false

解决方法:

- concurrency: 10
- max_bulk_action: 2000
- 1.5亿(父子结构)数据 一台4c 16g的es机器   8小时导入完成



```
<Regex>                                                                          <Num
▼ org.jboss.netty.channel.socket.nio.NioAcceptedSocketChannel @ 0x73be54d30
    ▶ org.jboss.netty.channel.socket.nio.AbstractNioChannel$WriteRequestQueue @ 0x73b
    ▶ org.jboss.netty.channel.DownstreamMessageEvent @ 0x716d94d40
    ▶ org.jboss.netty.channel.socket.nio.SocketSendBufferPool$UnpooledSendBuffer @ 0x
    ▶ org.jboss.netty.channel.DefaultChannelPipeline @ 0x73be54da8
    ▶ org.jboss.netty.channel.AbstractChannel$ChannelCloseFuture @ 0x73be54f40
    ▶ org.jboss.netty.channel.socket.nio.DefaultNioSocketChannelConfig @ 0x73be55290
    ▶ java.net.InetSocketAddress @ 0x73be55008
    ▶ java.net.InetSocketAddress @ 0x73be55060
    ▶ org.jboss.netty.channel.SucceededChannelFuture @ 0x73be54f30
    ▶ java.lang.Object @ 0x73be54f70
    ▶ org.jboss.netty.channel.socket.nio.AbstractNioChannel$WriteTask @ 0x73be54f80
    ▶ java.util.concurrent.atomic.AtomicBoolean @ 0x73be54f90
    ▶ java.util.concurrent.atomic.AtomicInteger @ 0x73be54fe8
    ▶ java.util.concurrent.atomic.AtomicInteger @ 0x73be54ff8
    ∑ Total: 14 entries
▶ org.jboss.netty.channel.socket.nio.NioAcceptedSocketChannel @ 0x73b6fefb0
▶ org.jboss.netty.channel.socket.nio.NioAcceptedSocketChannel @ 0x7455ff090
▶ org.elasticsearch.transport.TransportService @ 0x6f206e900
```

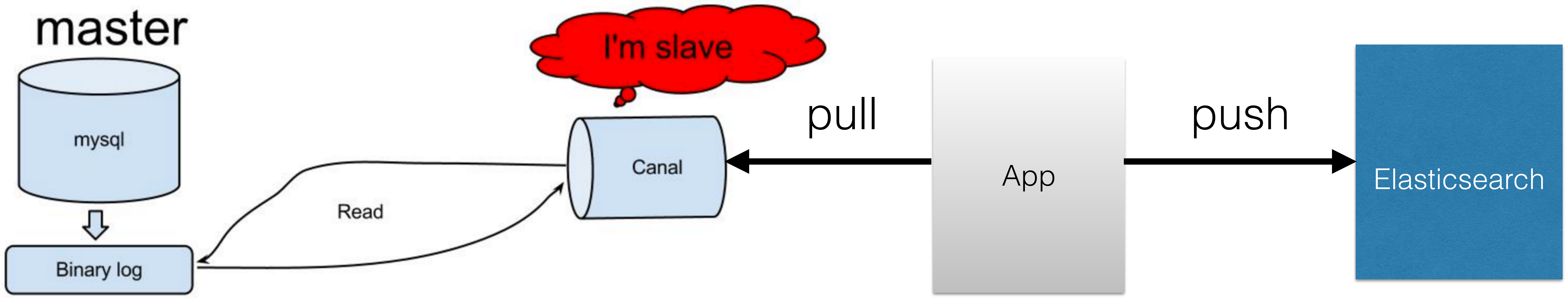
Incremental data

```
⊟{
    "type":"jdbc",
    "schedule":"0 */10 * * * ?",
    "jdbc":⊟{
        "url":"jdbc:mysql://localhost:3306/test",
        "statefile":"statefile.json",
        "user":"",
        "password":"",
        "sql":⊟[
            ⊟{
                "statement":"select * from products where mytimestamp > ?",
                "parameter":⊟[
                    "$metrics.lastexecutionstart"
                ]
            }
        ],
        "index":"my_jdbc_index",
        "type":"my_jdbc_type"
    }
}
```

缺点

- 定时轮询
- 多数据源配置麻烦
- sql的时间条件不准确

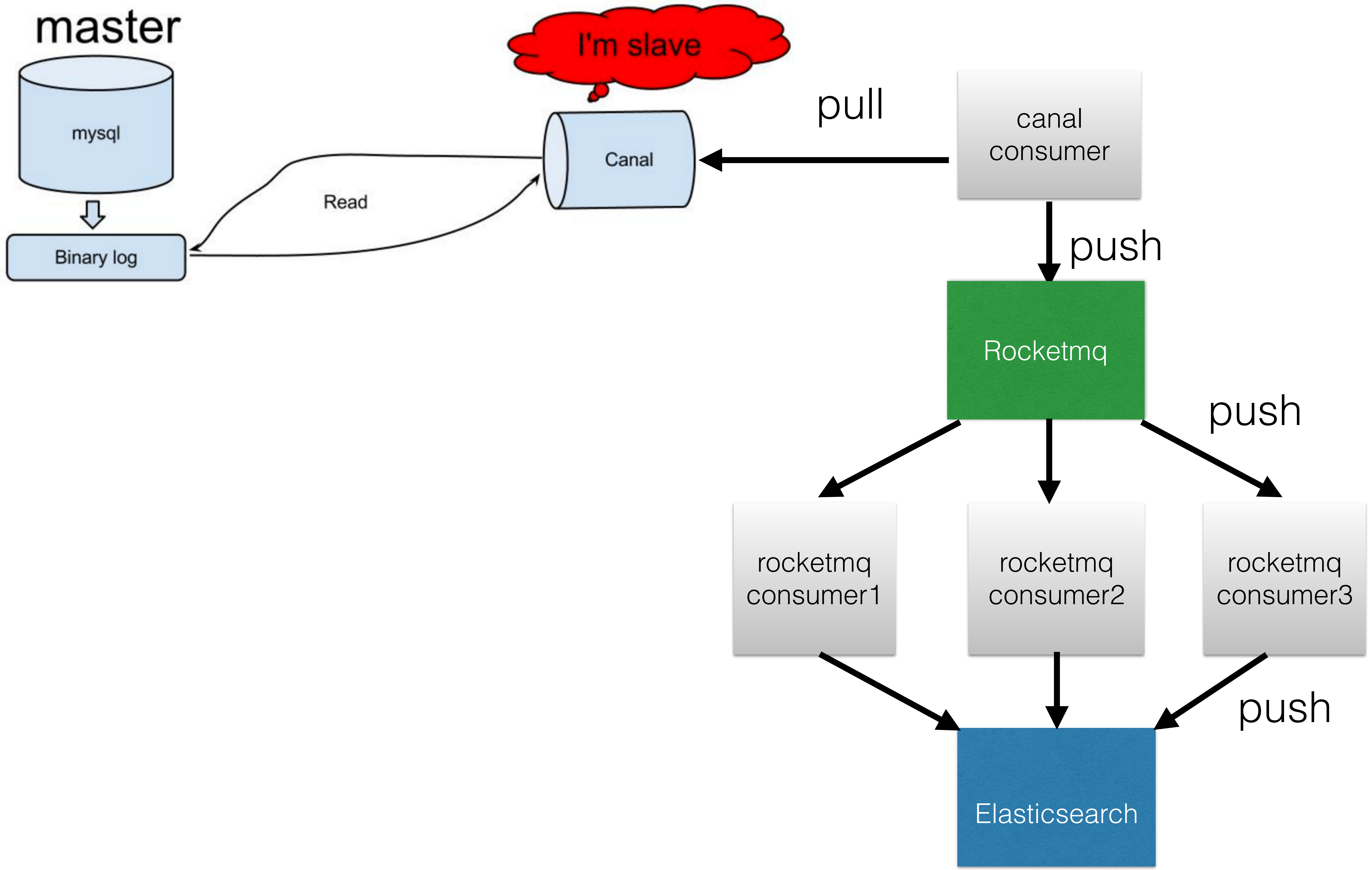
Mysql binlog


master
mysql
Binary log
Read
I'm slave
Canal
pull
App
push
Elasticsearch

we need more

- 回溯消费binlog
- binlog的广播
- 定时场景
- binlog的有序消费

Rocketmq


master
mysql
Binary log
Read
I'm slave
Canal
pull
canal consumer
push
Rocketmq
push
rocketmq consumer1
rocketmq consumer2
rocketmq consumer3
push
Elasticsearch

- 最好不要用mq的顺序功能
- version控制数据的并发＋ 消费幂等
- 定期做一次全量 (alias不停机迁移)

future

- canal 对 mysql vip的支持
- mysql ——> hadoop
- mongodb ?
- 监控
- 负载均衡节点和数据节点的隔离、读写的分离、集群的隔离

“欢迎加入码耘网络，共创跨境电商蓝图。”

– @卢栋