



ENJOY SAFER
TECHNOLOGY™

Using Apache Spark for generating ElasticSearch indices offline

Apache: Big Data Europe 2016

Andrej Babolčai
ESET Database systems engineer

Who am I

- Software engineer in database systems team
- Responsible for collecting, moving and providing access to data

Context



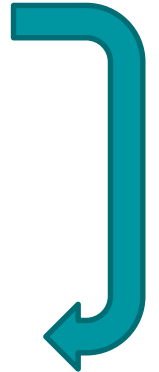
Apache
Kafka



Apache
Hive/Impala



ElasticSearch



Agenda

- Approaches we tried and why they failed
- Solution used, Spark + ES
- Benchmark, summary and possible improvements

Agenda

- Approaches we tried and why they failed
- Solution used, Spark + ES
- Benchmark, summary and possible improvements

Indexing data to live cluster

- Failed because of
 - Slowed search and near real-time (NRT) import
 - Reduce ingestion speed - too slow

Spark job with Lucene library

- **Approach**
 - Generate indices with Lucene and “import” them to ES
 - Indexing with Lucene is fast, hundreds of GB/hour
- **Failed because of**
 - ES types in Lucene
 - ES translog and checksum

Agenda

- Approaches tried and why they failed
- Solution used, Spark + ES
- Benchmark, summary and possible improvements

Goal

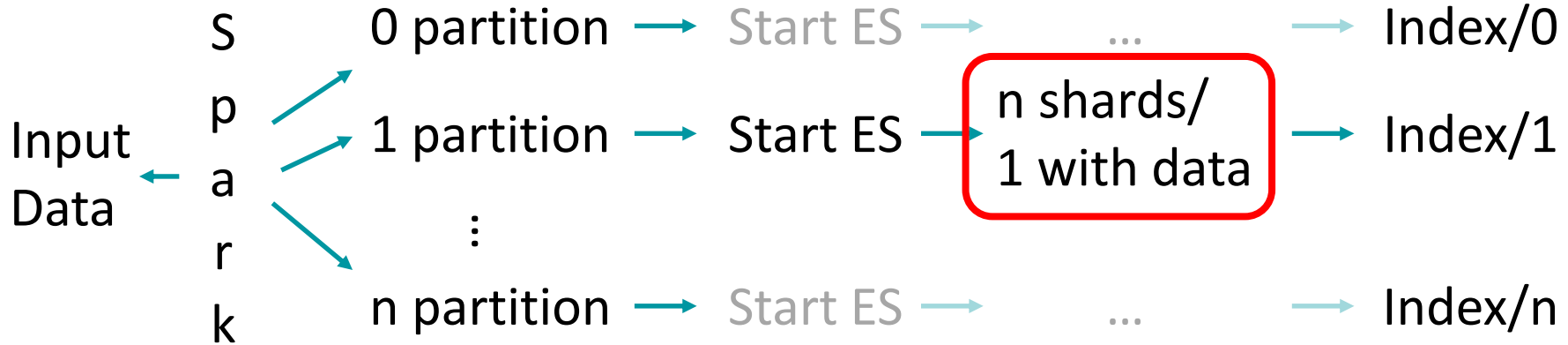
- Offload ES cluster and generate indices on Spark cluster
- We want indices to be “ready to use”
- When appropriate copy them to ES

Spark + local ES

- Based on <https://github.com/MyPureCloud/elasticsearch-lambda>
- Similar approach to Cloudera Solr MapReduceIndexerTool

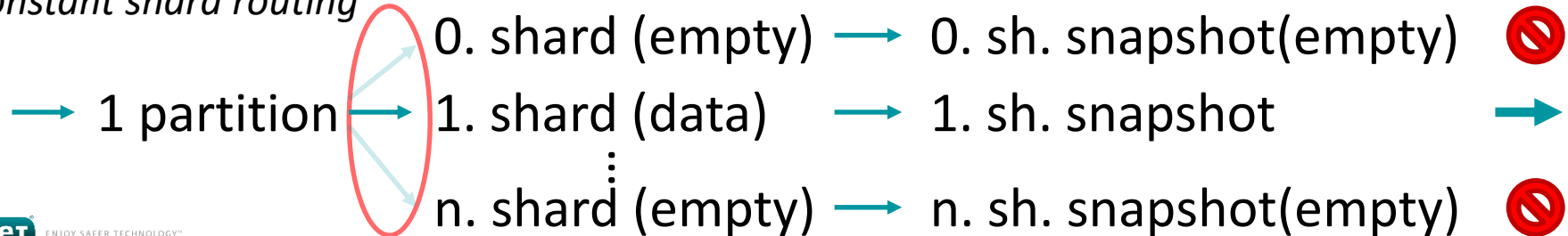
How do we generate indices offline

HDFS repository

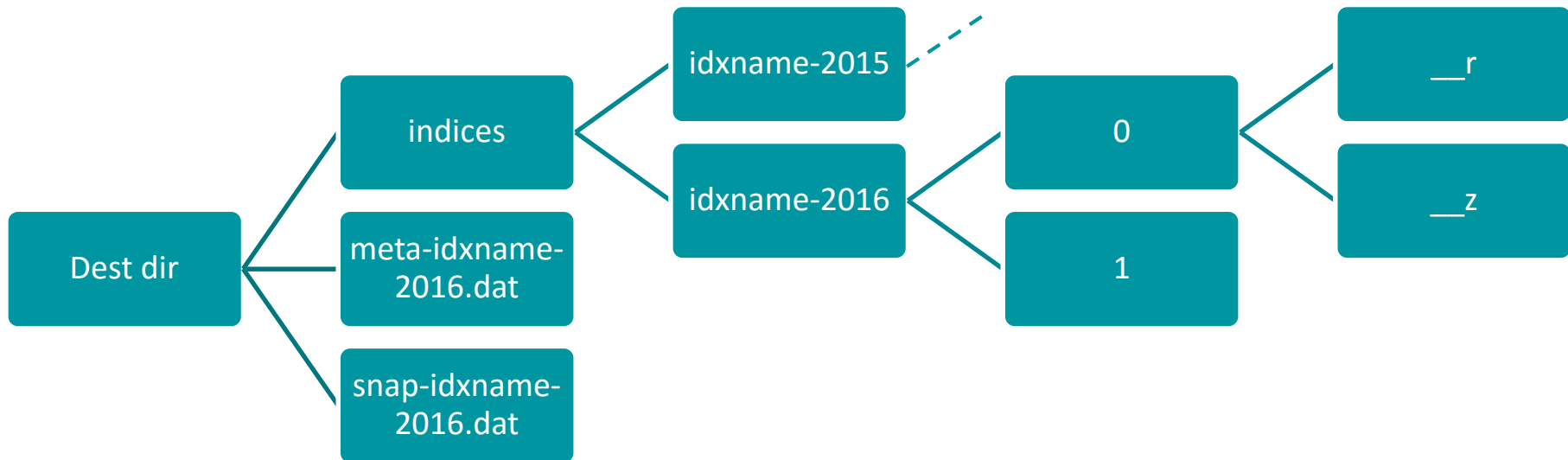


Snapshot

Constant shard routing



HDFS snapshot repository layout



Creating local ES node

```
val nodeSettings: Settings = Settings.builder
    .put("http.enabled", false)
    .put("processors", 1)
    .put("index.merge.scheduler.max_thread_count", 1)
    ...
    .build
```

Only JVM local node discovery



```
val node: Node =
    nodeBuilder().settings(nodeSettings).local(true).node()
node.start
val client: Client = node.client
client.admin.indices. ... .setSource(mapping).get
```





Same json mapping as Index API (http)

RDD export like *saveAsTextFile*

```
rddToIndex  
  .repartition(config.numShards)  
  .saveToESSnapshot(  
    config,...
```

We use implicit conversions

```
package object spark {  Infiltrate spark namespace  
  implicit class  
    DBSysSparkRDDFunctions  
      [T <: Map[String, Object]] Input RDD type bound //Row  
        (val rdd: RDD[T]) extends AnyVal {  
  def saveToESSnapshot(config:String,...):Unit = {  
  ...  
  rdd.sparkContext.runJob(rdd, esWriter.processPartition _)}
```

 *Indexing method*

Useful ES commands

- Create HDFS snapshot repository:

```
curl -s -XPUT 'localhost:9200/_snapshot/<Repo name>' -d '{
  "type": "hdfs",
  "settings": {
    "uri": "hdfs://namenode:8020/",
    "path": "/user/<username>/<Snapshot repo hdfs path>",
    "load_defaults": "false"
  }
}'
```


Useful ES commands

- Start restore process:

```
curl -XPOST 'localhost:9200/_snapshot/<Repo name>/<snapshot name>/_restore'
```

- Monitor restore progress:

```
curl -s -XGET 'localhost:9200/_cat/recovery?v' |  
awk '{print $1 " " " $11}' |  
fgrep -v " 0.0%" |  
fgrep -v "100.0%"
```

Agenda

- Approaches tried and why they failed
- Solution used, Spark + ES
- Benchmark, summary and possible improvements

ES cluster configuration

Property	Value
Number of nodes	24
ES heap size	29GB
CPUs	8 (/proc/cpuinfo)
HDD	2x3.5TB / node
No. of indices	130
No. of shards	~3900
Data size	16 TB
No. of docs	> 60 billion
Indexing speed (what we can handle...)	~1000 docs/s

Offline indexing environment

Property	Value
Input size	135GB compr. parquet
Number of docs	470M
CPUs (indexing)	15 spark workers
Memory	4GB /worker
Output index layout	20 string fields, 15 part.
Job duration	~3.5h
Restore duration	~20m
Duration total	~4h
Indexing speed	>30k docs/s

Future work

- Shard routing
- Indexing on local FS, use directly HDFS
- Speed up indexing
- Use for stream indexing

Summary

- Hard to directly compare RT with our offline approach
- What we wanted was to make historical data available for users, without influencing production systems

<https://github.com/andybab/OfflineESIndexGenerator>



ENJOY SAFER
TECHNOLOGY™

Thank you
Questions?

babolcai@eset.sk