elasticsearch data/

About Me

- Igor Motov
- Developer at Elasticsearch Inc.
- Github: imotov
- Twitter: @imotov

About Elasticsearch Inc.

- Founded in 2012

  By the people behind the Elasticsearch and Apache Lucene
  http://www.elasticsearch.com
  Headquarters: Amsterdam and Los Altos, CA

- We provide

  Training (public & onsite)
  Development support
  Production support subscription (SLA)

file descriptors

"Make sure to increase the number of open files descriptors on the machine (or for the user running elasticsearch). Setting it to 32k or even 64k is recommended."

Source: setup and configuration guide

where are all these file descriptors go?

files, data structures and their usage

main concepts

- node

  a running elasticsearch instance (typically JVM process)

- cluster

  a group of nodes sharing the same set of indices

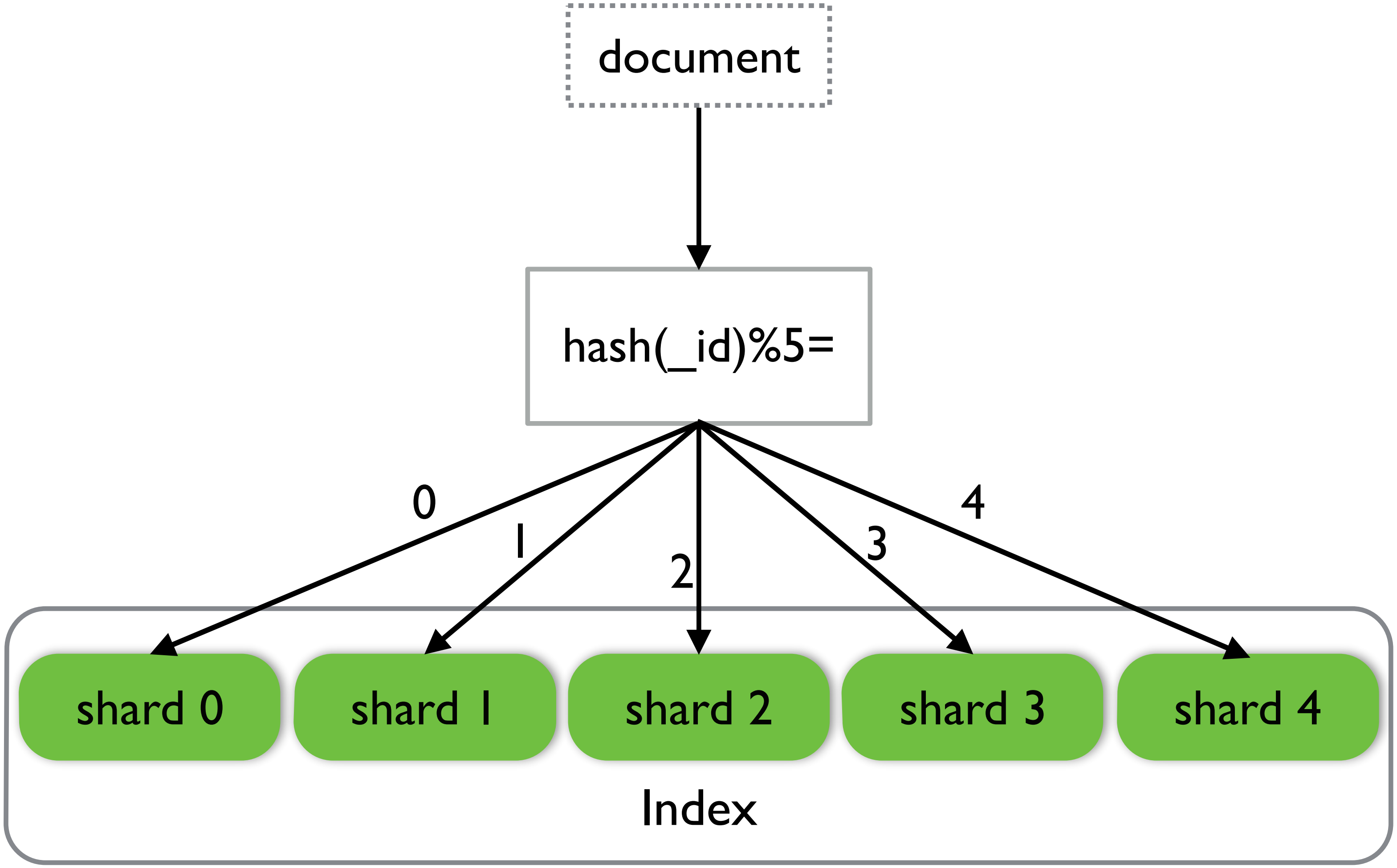
- index

  a set of documents of possibly different types
  stored in one or more shards

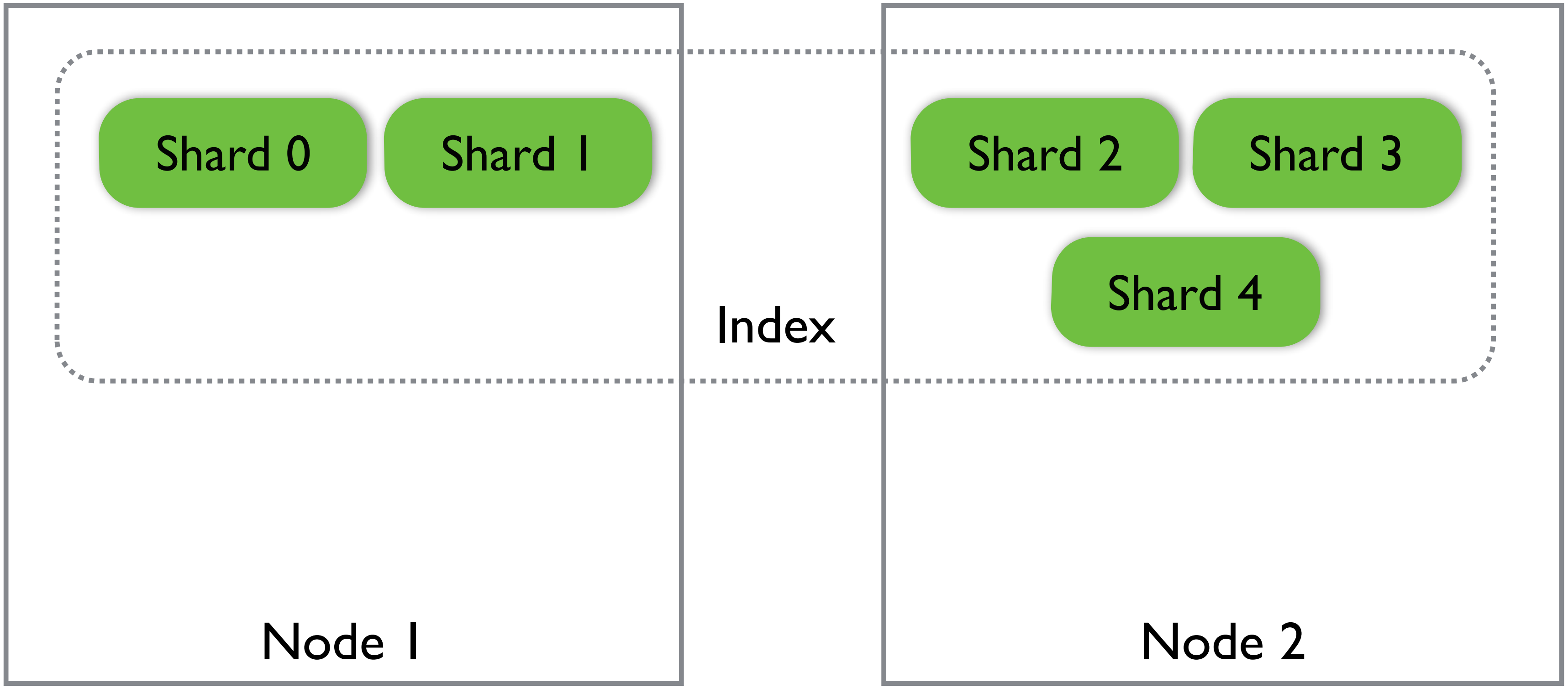
- shard

  a lucene index, allocated on one of the nodes

shards


document
hash(_id)%5=
0
1
2
3
4
shard 0
shard 1
shard 2
shard 3
shard 4
Index

shards



master node

- elected when nodes form a cluster
- coordinates work of other nodes through cluster state
- the only node that can update cluster state
- publishes cluster state to other node

cluster state

- nodes

  list of nodes in the cluster, their addresses, attributes and master

- index metadata

  settings, mappings and aliases

- shard routing table

  where the shards can be found

- index templates

- cluster settings

  persistent and transient

cluster state - persistent

- nodes

  list of nodes in the cluster, their addresses, attributes and master

- index metadata

  settings, mappings and aliases

- shard routing table

  where the shards can be found

- **index templates**

- cluster settings

  persistent and transient

data

- node level

  persistent cluster settings, templates

- index level

  aliases, index settings, mappings

- shard level

  shard metadata, lucene index, transaction log

data directory

- “data” directory in elasticsearch home by default
- `path.data` in config/elasticearch.yml
- --`path.data=`... on command line
- handled by deb and rpm packages

multiple nodes per data dir

- <data_dir>/<cluster_name>/nodes/NNN
  where NNN = 0, 1, 2, ...

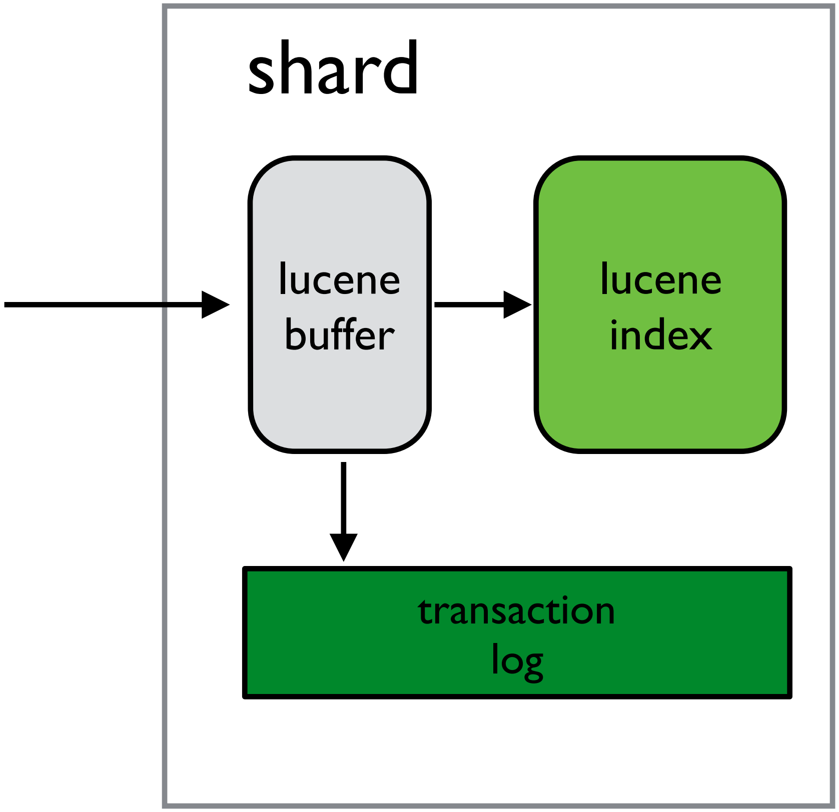
- `node.max_local_storage_nodes`
  default 50

let’s take a look

summary

```
<cluster>/
    nodes/
        <N>/
            _state/ - cluster state
            node.lock - lock
            indices/
                <index-name>/
                    _state/ - index metadata
                    0/
                        _state/ - shard metadata
                        index/ - index data
                        translog/ - transaction log data
```

transaction log



transaction log

- transaction log

  stores every operation (create/update/delete)
  fsync-ed every 5 sec (configurable)
  replayed on node restart

- lucene segments

  fsync-ed when transaction log is full (every 30 min, 200mb or 500 operations)

lucene index

- inverted index
- stored fields
- doc values
- …

inverted index

- Document 1:

```
{
   “text”: “Elasticsearch is an open source, distributed search
engine.”,
   “date”: “2014-07-01”
}
```

- Document 2:

```
{
   “text”: “Elasticsearch is a search server based on Lucene.”,
   “date”: “2014-07-02”
}
```

analysis

- “Elasticsearch is an open source, distributed search engine.” could be translated into tokens:
  - elasticsearch
  - open
  - source
  - distributed
  - search
  - engine

- “Elasticsearch is a search server based on Lucene.” could be translated into tokens:
  - elasticsearch
  - search
  - server
  - based
  - lucene

inverted index - field text

token	document frequency	postings (document ids)
based	1	2
distributed	1	1
elasticsearch	2	1, 2
engine	1	1
lucene	1	2
open	1	1
search	2	1, 2
server	1	2
source	1	1

inverted index - field date

token	document frequency	postings (document ids)
2014-07-01	1	1
2014-07-02	1	2

inverted index

- tokens->documents
- easy to build
- difficult to update
- segmented
- segments are merged periodically

field data

- “uninverted" inverted index
- documents->tokens
- can be built from inverted index on demand
- can be stored with index as doc values
- segmented
- used by sorting, aggregations, scripts, etc

field data - text

document	tokens
1	*distributed, elasticsearch, engine, open, search, source*
2	*based, elasticsearch, lucene, search, server*

field data - date

document	tokens
1	*2014-07-01*
2	*2014-07-02*

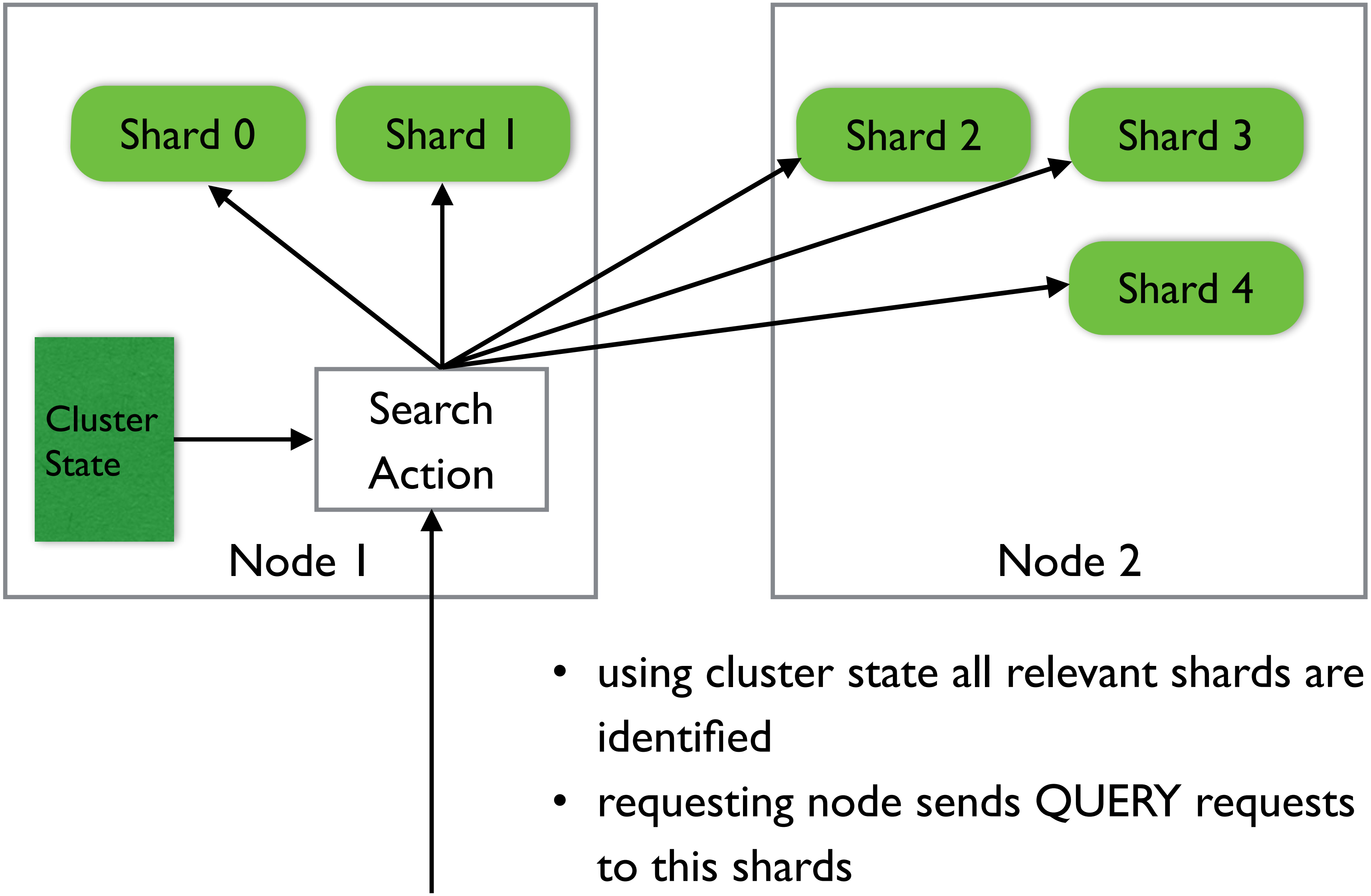
stored fields

- _source - JSON source of the entire document
- _parent id
- routing
- ttl
- _uid
- any other field marked as "stored"

all together now

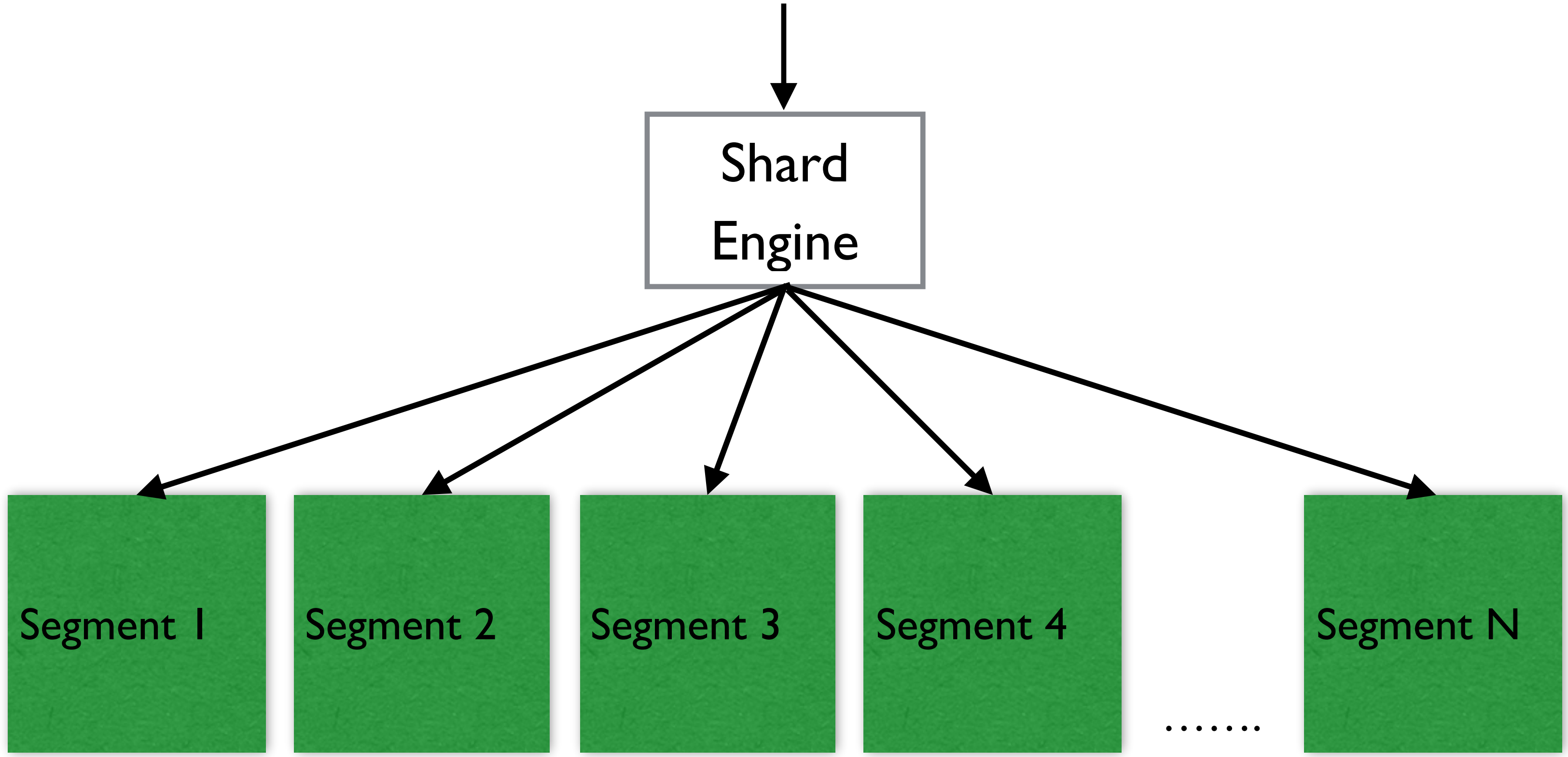
- searching for terms “distributed” and “service”
- sorting by the field “date”

QUERY phase - node level



- using cluster state all relevant shards are identified
- requesting node sends QUERY requests to this shards

QUERY phase - shard level



- each shard searches all segments in the shard one after another

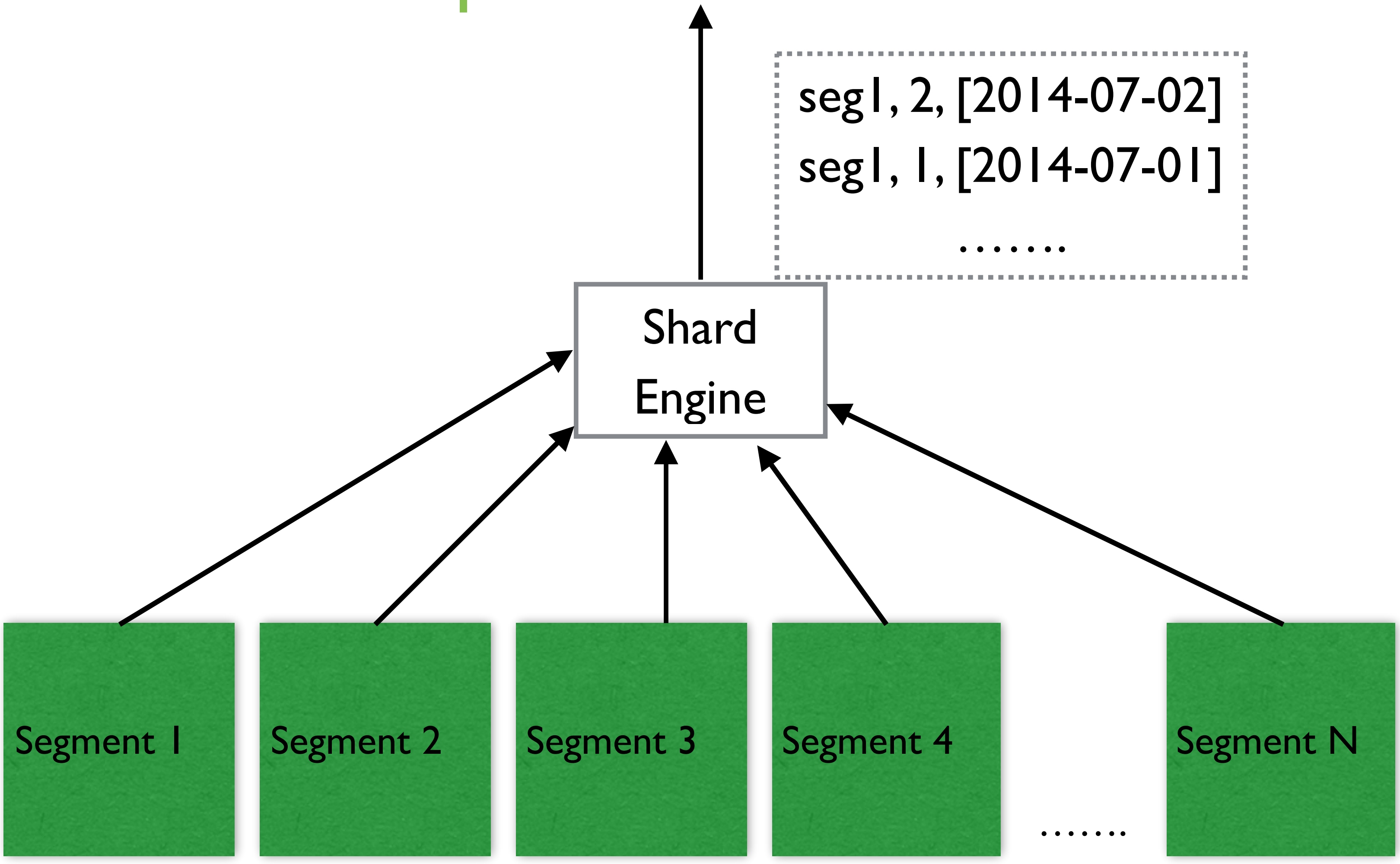
QUERY phase - inverted index

token	document frequency	postings (document ids)
based	1	2
distributed	1	**1**
elasticsearch	2	1, 2
engine	1	1
lucene	1	2
open	1	1
search	2	1, 2
server	1	**2**
source	1	1

QUERY phase - field data

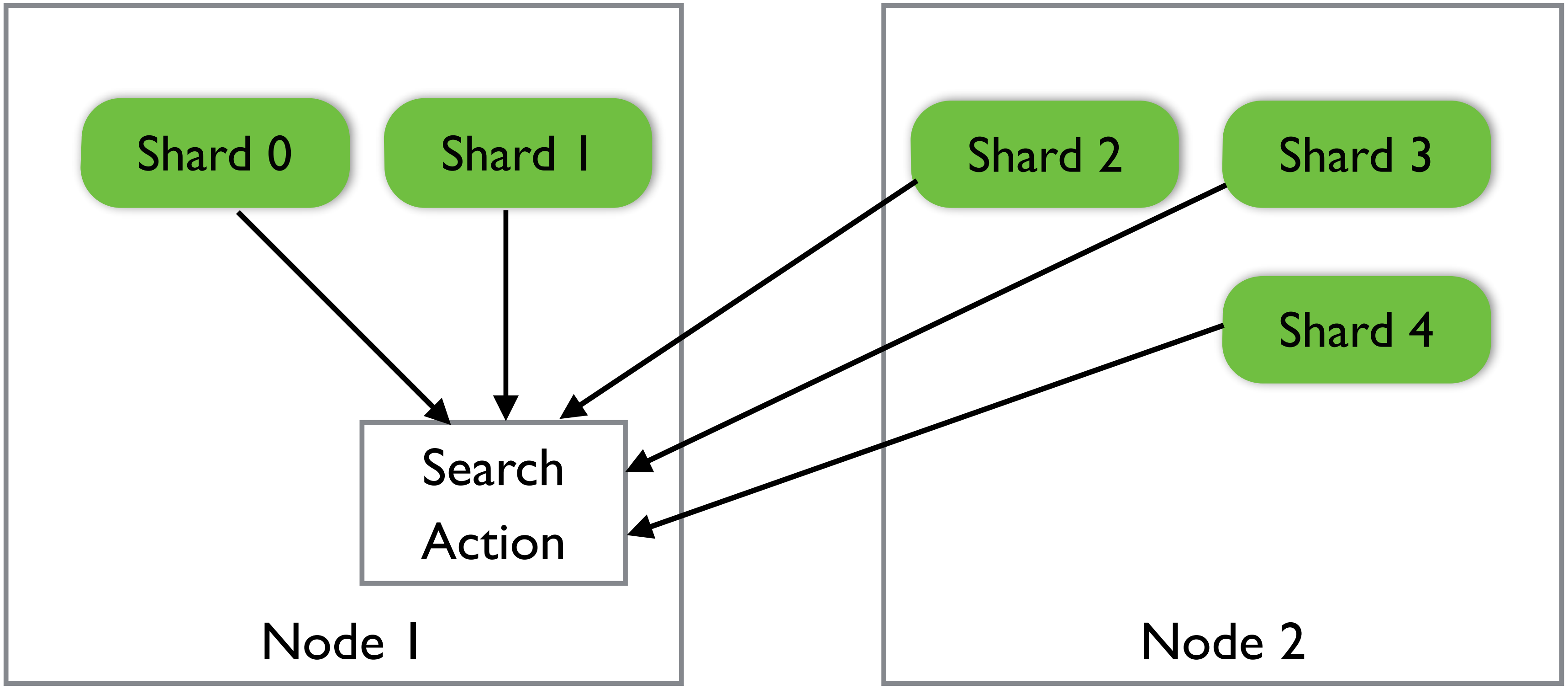
document	tokens
1	***2014-07-01***
2	***2014-07-02***

QUERY phase - shard level



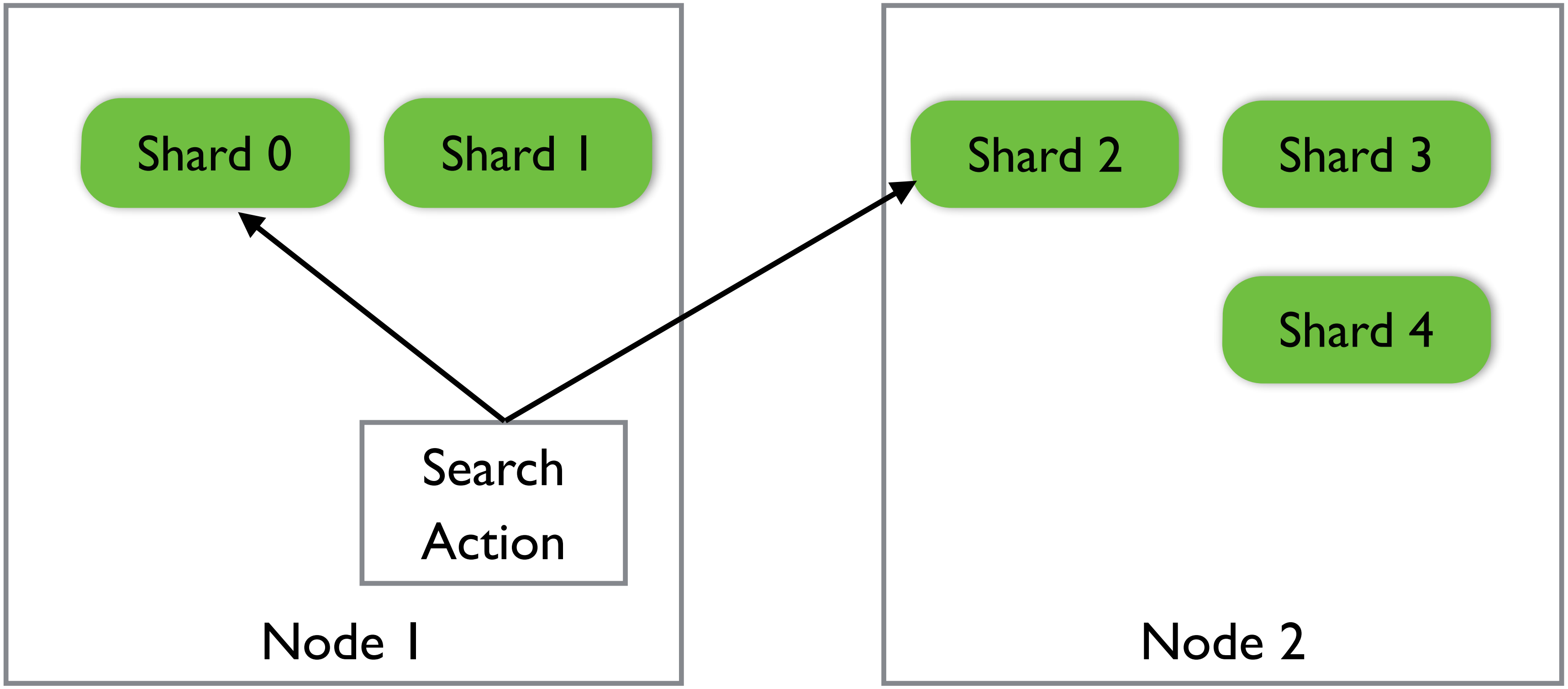
- all segments are searched and top 10 documents are collected for each shard
- for each document internal Lucene id and sort key is stored

QUERY phase - node level



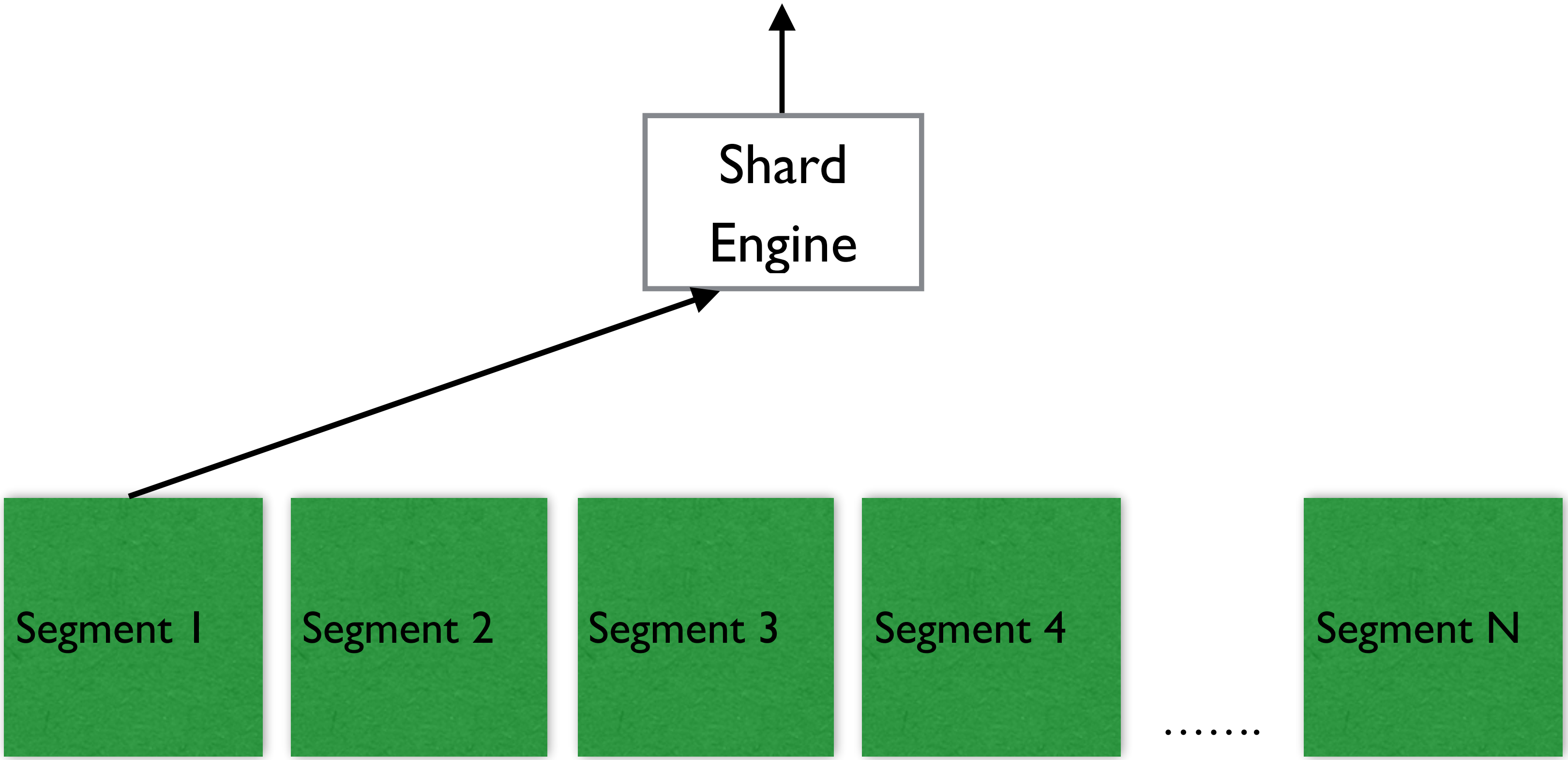
- top 10 ids and sort keys for each shard are sent to requesting node
- requesting node resorts them and finds global top10

FETCH phase - node level



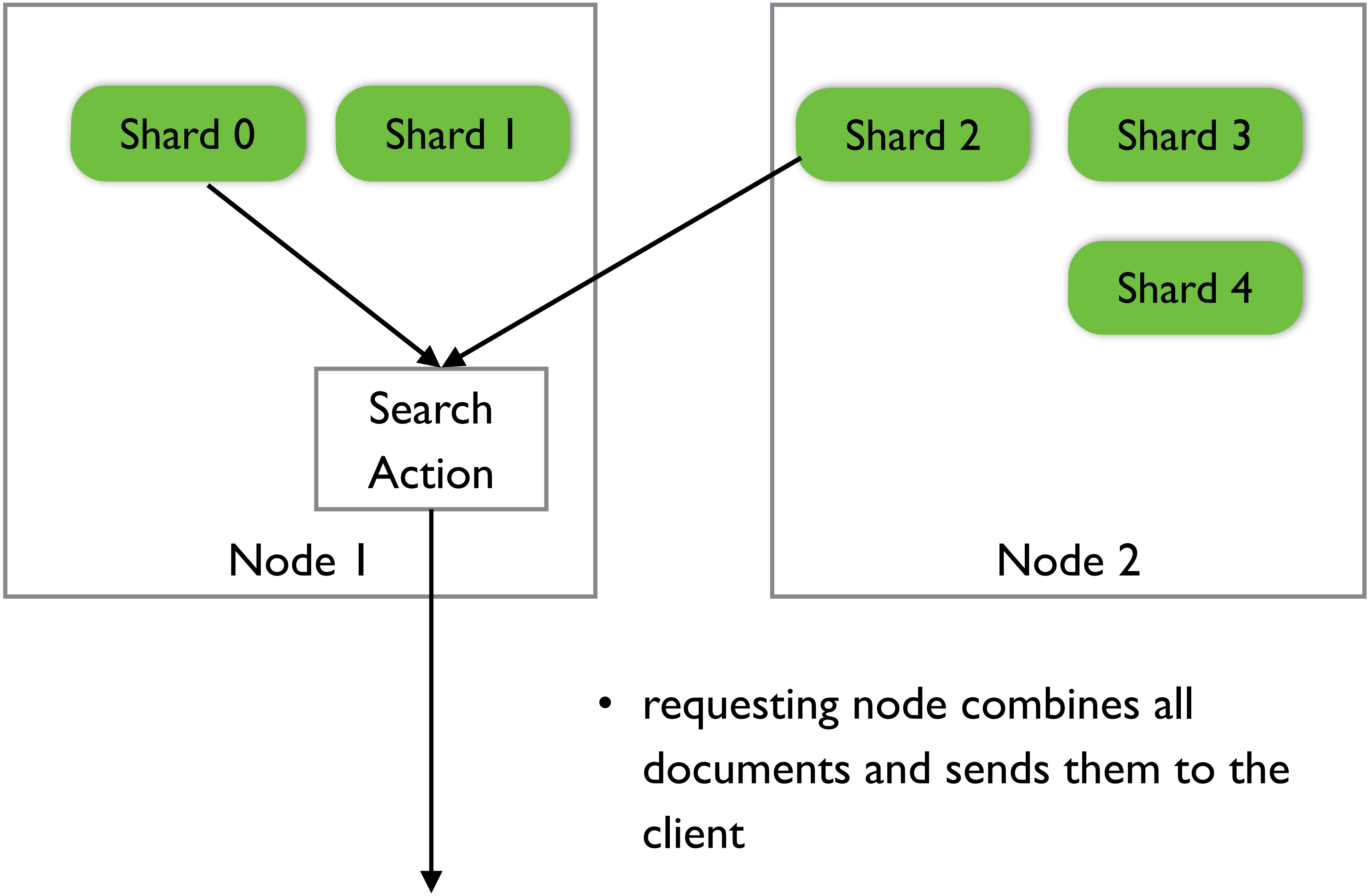
- global top 10 documents are requested
- only shards that have these top 10 documents are contacted

FETCH phase - shard level



- _source (stored field) is retrieved from corresponding segments

FETCH phase - node level



- requesting node combines all documents and sends them to the client

elasticsearch.

… and this is it

questions?