CENTERDEVICE

codecentric
agile software factory

The Do's and Don’ts of
Elasticsearch Scalability and Performance

Patrick Peschlow

Think hard about your mapping

Think hard about your mapping

- Which fields to analyze? How to analyze them?
- Need term frequencies, positions, offsets? Field norms?
- Which fields to not analyze or not index/enable?
- _all
- _source vs stored fields

Think hard about your mapping

- Dynamic mapping/templates
  - Excessive number of fields?

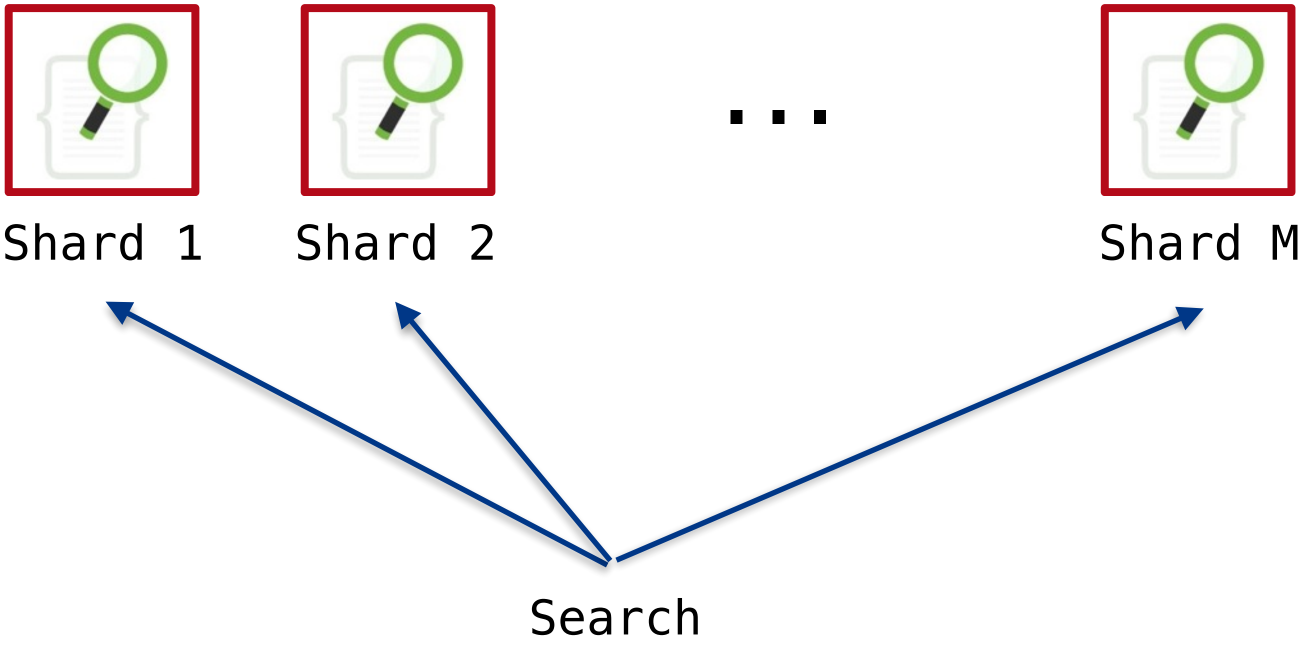
- Index-time vs. query-time solutions

- Multi field, copy to, transform script

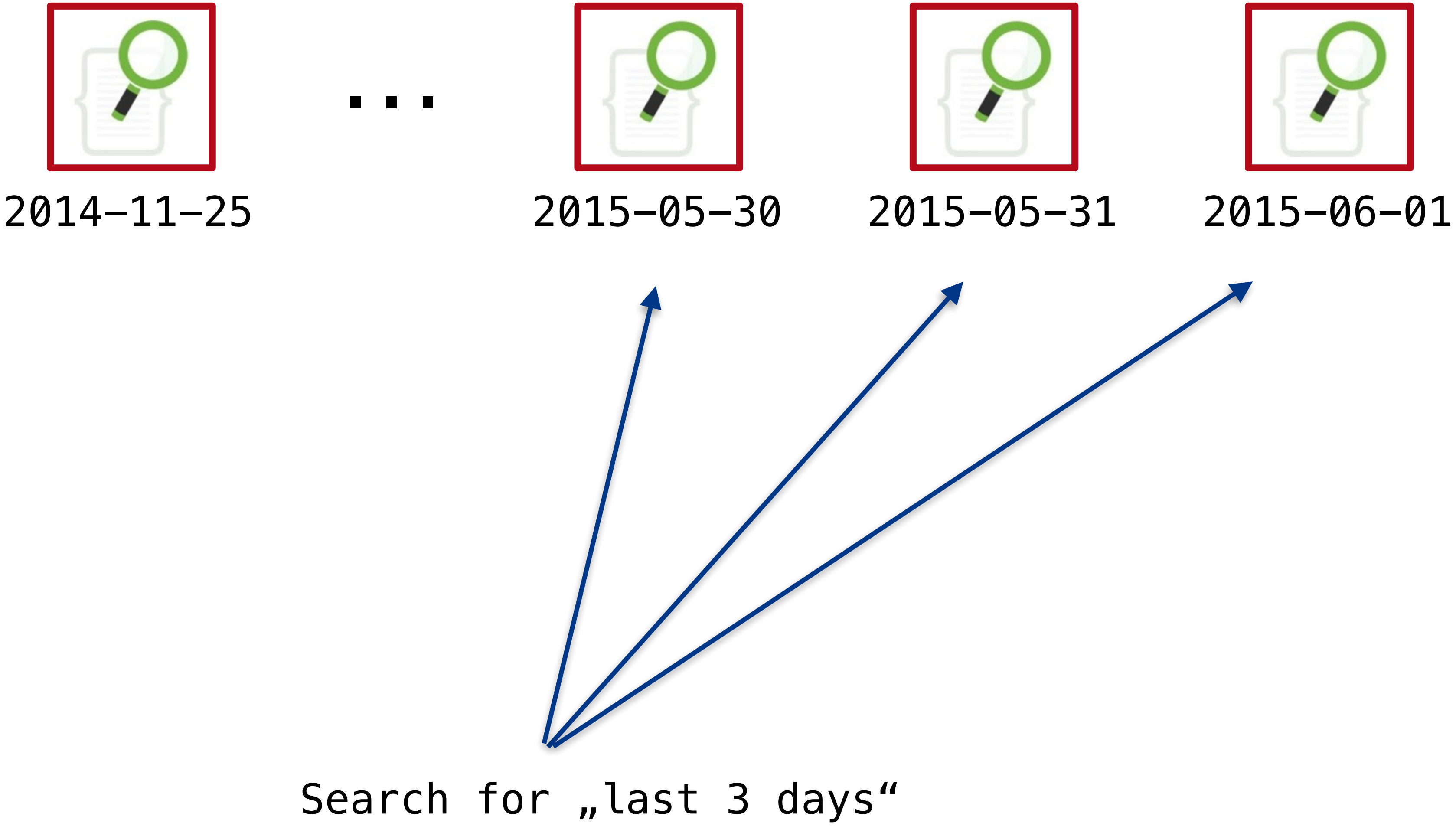
- Relations: parent-child/nested

Design for scale

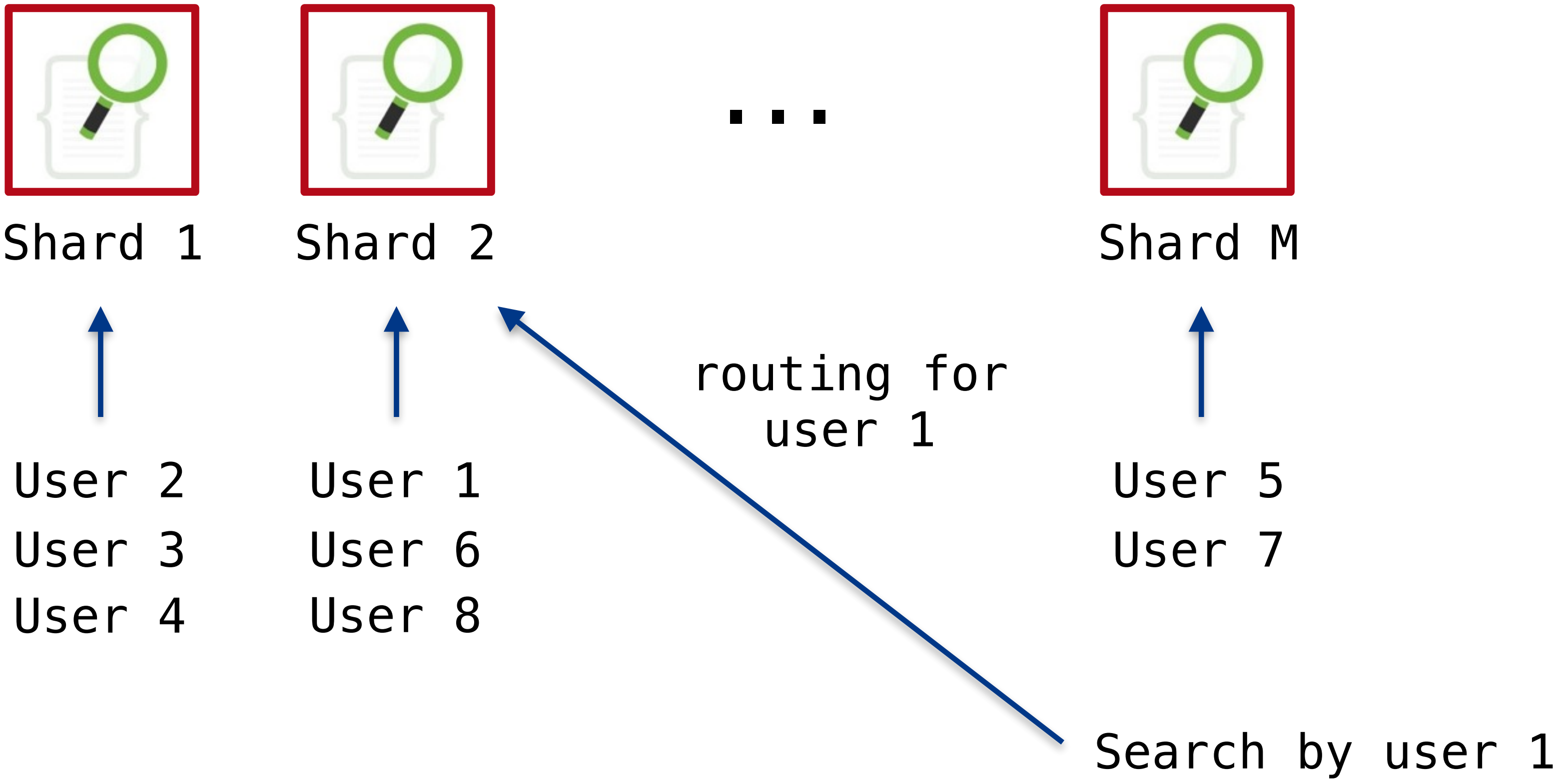
Design for scale



Design for scale



Design for scale



Shard 1

Shard 2

...

Shard M

User 2
User 3
User 4

User 1
User 6
User 8

routing for
user 1

User 5
User 7

Search by user 1

Design for scale

- Can documents/access be partitioned in a natural way?
- Need to find documents by ID (update/delete/get)?
- Know the relevant features
    - Routing, aliases, multi-index search
- Indices don't come for free
- Measure the impact of distributed search

Don’t create more shards than you need

Don’t create more shards than you need

- More shards
  - Enable larger indices
  - Scale operations on individual documents

- But shards don’t come for free

- Measure how many shards you need
  - When unsure, overallocate a little

✗

Don’t treat all nodes as equal

Don’t treat all nodes as equal

- Cluster nodes
  - Master nodes, data nodes, client/aggregator nodes

- Client applications
  - HTTP?
  - Transport protocol?
  - Join the cluster as a client node?
  - In Java: HTTP client vs TransportClient vs NodeClient

✗

Don’t run wasteful queries

Don’t run wasteful queries

- Only request as many hits as you need
- Avoid deep pagination
- Use scan+scroll to iterate without sorting
- Only query indices/shards that may contain hits

Engineer queries

Engineer queries

- Measure performance
  - Set up production-like cluster and data
- Use filters
- Check and tune filter caching
- Reduce work for heavyweight filters
  - Order them, consider accelerators

✓

Care about field data

Care about field data

- Used for sorting, aggregation, parent-child, scripts, …
- High memory consumption or OutOfMemoryError
  - Cache limit, circuit breakers avoid the worst
- Evaluate field data requirements in advance
- Use „doc values“ to store expensive field data on disk

Be prepared
for reindexing

Be prepared for reindexing

- Reasons for reindexing
  - Mapping changes
  - Index/shard reaches its capacity
  - Reduce number of indices/shards

Be prepared for reindexing

- Reindexing procedure depends on many factors
  - Data source?
  - Zero downtime?
  - Update API usage?
  - Possible deletes?
  - Designated component (queue) for indexing?

Be prepared for reindexing

- Use existing tooling

- Do it yourself? Use scan+scroll and bulk indexing

- Follow best practices
  - Use aliases
  - Disable refresh
  - Decrease number of replicas

✗ Don’t use the defaults

Don’t use the defaults

- Cluster settings
  - cluster name, discovery, minimum_master_nodes
  - recovery
- Number of shards and replicas
- Refresh interval
- Thread pool and cache configuration

Monitor

Monitor

- Cluster health, split brains
- Thread pools and caches
- Garbage collection (the actual JVM output)
- Slow log
- Hot threads

Follow the production recommendations

Follow the production recommendations

- A good start would be to read/research them at all

- Just to mention a few
  - The more memory, the better
  - Isolate as much as possible
  - SSDs and local storage recommended

Don’t test in production

Don’t test in production

- Use a test environment

- Test the cluster
  - Single node restarts, rolling upgrades, node loss
  - Full cluster restarts

- Test behavior under expected load
  - Queries
  - Indexing

Read the guide

Read the guide

- Elasticsearch: The Definitive Guide
  - https://www.elastic.co/guide/en/elasticsearch/guide/current/index.html
- Elasticsearch Reference
  - https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html

Questions?

Dr. rer. nat. Patrick Peschlow

codecentric AG
Merscheider Straße 1
42699 Solingen

tel +49 (0) 212.23 36 28 54
fax +49 (0) 212.23 36 28 79
patrick.peschlow@codecentric.de

www.codecentric.de