



elasticsearch

# (R)Evolution

Philipp Krenn

@xeraa





# Revolution

343 systems in ranking, June 2018

Rank			DBMS	Database Model	Score		
Jun 2018	May 2018	Jun 2017			Jun 2018	May 2018	Jun 2017
1.	1.	1.	Oracle <span>+</span>	Relational DBMS	1311.25	+20.84	-40.51
2.	2.	2.	MySQL <span>+</span>	Relational DBMS	1233.69	+10.35	-111.62
3.	3.	3.	Microsoft SQL Server <span>+</span>	Relational DBMS	1087.73	+1.89	-111.23
4.	4.	4.	PostgreSQL <span>+</span>	Relational DBMS	410.67	+9.77	+42.13
5.	5.	5.	MongoDB <span>+</span>	Document store	343.79	+1.67	+8.79
6.	5.	5.	DB2 <span>+</span>	Relational DBMS	85.51	+0.00	-1.86
7.	7.	<span>↑</span> 9.	Redis <span>+</span>	Key-value store	136.30	+0.95	+17.42
8.	<span>↑</span> 9.	<span>↑</span> 11.	Elasticsearch <span>+</span>	Search engine	131.04	+0.60	+19.48
9.	<span>↓</span> 8.	<span>↓</span> 7.	Microsoft Access	Relational DBMS	130.99	-2.12	+4.44
10.	10.	<span>↓</span> 8.	Cassandra <span>+</span>	Wide column store	119.21	+1.38	-4.91
11.	11.	<span>↓</span> 10.	SQLite <span>+</span>	Relational DBMS	114.26	-1.19	-2.44
12.	12.	12.	Teradata	Relational DBMS	75.77	+1.36	-1.55
13.	<span>↑</span> 14.	<span>↑</span> 18.	MariaDB <span>+</span>	Relational DBMS	65.85	+0.85	+12.95
14.	<span>↓</span> 13.	<span>↑</span> 16.	Splunk	Search engine	65.78	+0.68	+8.26
15.	15.	<span>↓</span> 14.	Solr	Search engine	62.06	+0.55	-1.55

<https://db-engines.com/en/ranking>

343 systems in ranking, June 2018

Rank			DBMS	Database Model	Score		
Jun 2018	May 2018	Jun 2017			Jun 2018	May 2018	Jun 2017
1.	1.	1.	Oracle	Relational DBMS	1311.25	+20.84	-40.51
2.	2.	2.	MySQL	Relational DBMS	1233.69	+10.35	-111.62
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1087.73	+1.89	-111.23
4.	4.	4.	PostgreSQL	Relational DBMS	410.67	+9.77	+42.13
5.	5.	5.	MongoDB	Document store	343.79	+1.67	+8.79
6.	6.	6.	DB2	Relational DBMS	185.64	+0.03	-1.86
7.	7.	9.	Redis	Key-value store	136.30	+0.95	+17.42
8.	9.	11.	Elasticsearch	Search engine	131.04	+0.60	+19.48
9.	8.	7.	Microsoft Access	Relational DBMS	130.99	-2.12	+4.44
10.	10.	8.	Cassandra	Wide column store	119.21	+1.38	-4.91
11.	11.	10.	SQLite	Relational DBMS	114.26	-1.19	-2.44
12.	12.	12.	Teradata	Relational DBMS	75.77	+1.36	-1.55
13.	14.	18.	MariaDB	Relational DBMS	65.85	+0.85	+12.95
14.	13.	16.	Splunk	Search engine	65.78	+0.68	+8.26
15.	15.	14.	Solr	Search engine	62.06	+0.55	-1.55



MS SQL [MS SQL] "Query is too complex" (self.SQL)

submitted 5 days ago by adman29

Hey guys, I'm using MS Access 2016, and after writing out the SQL Statement to combine all of my tables, I'm getting an error stating "Query is too complex." Is there a work around for this?

**8 comments** share save hide report pocket

[https://www.reddit.com/r/SQL/comments/7i828i/ms\\_sql\\_query\\_is\\_too\\_complex/](https://www.reddit.com/r/SQL/comments/7i828i/ms_sql_query_is_too_complex/)

# Who uses Elasticsearch?



elastic

Infrastructure | Developer Advocate

[http://thedudeabides.com/articles/  
the\\_birth\\_of\\_compass](http://thedudeabides.com/articles/the_birth_of_compass)





elasticsearch.

# Terms

Cluster, Node, Index, Shard, Document, ID

```
$ curl http://localhost:9200
{
  "name": "elasticsearch1",
  "cluster_name": "docker-cluster",
  "cluster_uuid": "l6wfwv8XSniI_Fx6qqrcw",
  "version": {
    "number": "5.6.9",
    "build_hash": "877a590",
    "build_date": "2018-04-12T16:25:14.838Z",
    "build_snapshot": false,
    "lucene_version": "6.6.1"
  },
  "tagline": "You Know, for Search"
}
```

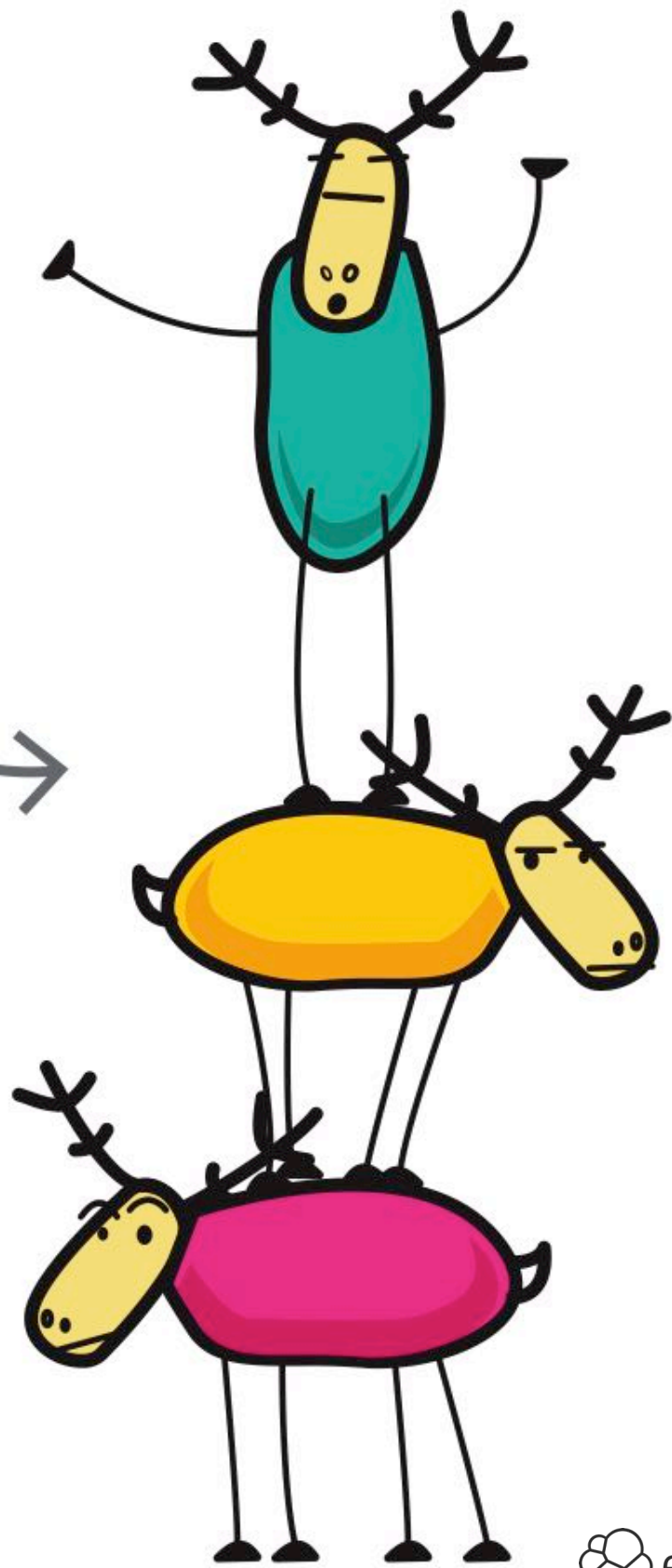
```
$ curl http://localhost:9200
{
  "name": "elasticsearch1",
  "cluster_name": "docker-cluster",
  "cluster_uuid": "16wfwv8XSniiI_Fx6qqrcw",
  "version": {
    "number": "5.6.9",
    "build_hash": "877a590",
    "build_date": "2018-04-12T16:25:14.838Z",
    "build_snapshot": false,
    "lucene_version": "6.6.1"
  },
  "tagline": "You Know, for Search"
}
```





**logstash**

ELK Stack!  
Get it?

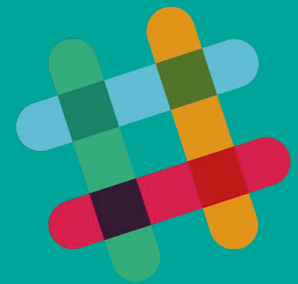


**E** Elasticsearch

**L** Logstash

**K** Kibana

mozilla



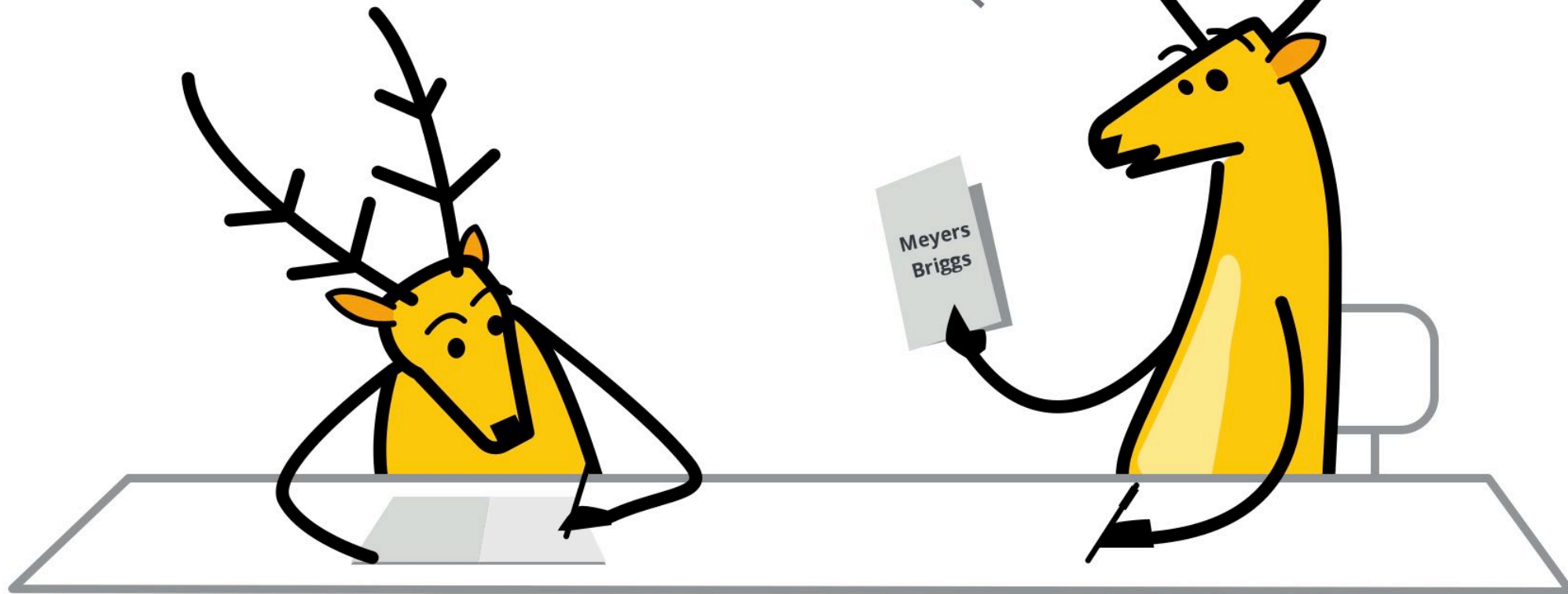
slack







*Apparently, I'm an  
ELKB personality.*







# elastic stack

# Evolution

Questions: <https://sli.do/xeraa>

Answers: <https://twitter.com/xeraa>



**I think you'll find I'm universally recognised as a mature and responsible adult.**

# Strictness\*

5.0

---

\* Demo



A black and white close-up portrait of a man with short hair, wearing dark sunglasses and a black leather jacket with silver studs. He has a slight, enigmatic smile. The background is a plain, light-colored wall.

IT'S BAD

# Bootstrap Checks

<https://www.elastic.co/guide/en/elasticsearch/reference/current/bootstrap-checks.html>

## Bootstrap Checks

Heap size check

File descriptor check

Memory lock check

Maximum number of threads check

Maximum size virtual memory  
check

Max file size check

Maximum map count check

Client JVM check

Use serial collector check

System call filter check

OnError and OnOutOfMemoryError  
checks

Early-access check

G1GC check

All permission check

# Parameters & Configs

# Rolling Upgrades\*

6.0

---

\* Demo



# Floodstage Watermark\*

6.0

---

\* Demo



**Low 85%**

**High 90%**

**Floodstage 95%**



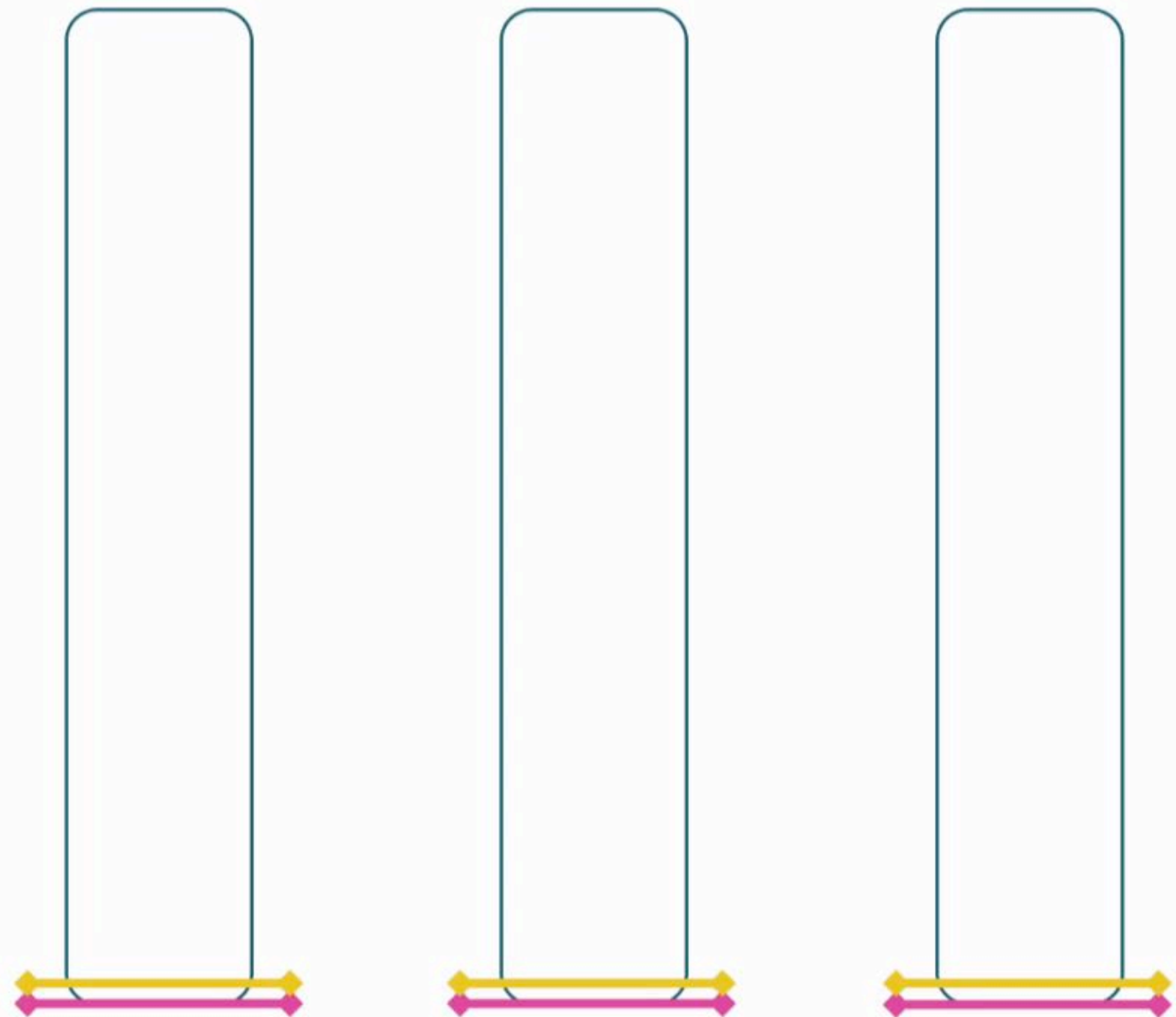
# Sequence Numbers\*

6.0

---

\* Demo





**Primary**  
*Term 1*

**Replica 1**

**Replica 2**

**Global checkpoint**  
**Local checkpoint**



63 bits ought to be  
enough for anyone.

# Tradeoff

```
index.translog.retention.size: 512MB
```

```
index.translog.retention.age: 12h
```

# Cross Datacenter Replication

6.x or 7.x

# ~~Types~~\*

## 5.6 to 8.0

---

\* Demo





# Why

Data types

Sparsity

Scoring

# How

**5.6** opt-in single type

**6.x** single type

**7.x** type optional in API

**8.x** no more types

[https://www.elastic.co/guide/en/elasticsearch/reference/current/removal-of-types.html#\\_schedule\\_for\\_removal\\_of\\_mapping\\_types](https://www.elastic.co/guide/en/elasticsearch/reference/current/removal-of-types.html#_schedule_for_removal_of_mapping_types)

# Automatic Queue Resizing

6.0

# Reject and Retry Instead of Long Queues

```
thread_pool.search.target_response_rate: 2s
```

Serving 50 requests/s

Queue size:  $2 * 50 = 100$

# Adaptive Replica Selection

6.1 (enabled by default in 7.0)



# **C3: Cutting Tail Latency in Cloud Data Stores via Adaptive Replica Selection**

*Lalith Suresh, Technische Universität Berlin; Marco Canini, Université catholique de Louvain;  
Stefan Schmid, Technische Universität Berlin and Telekom Innovation Labs;  
Anja Feldmann, Technische Universität Berlin*

<https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/suresh>

# Pick Best Shard

Exponentially Weighted Moving Average  
(EWMA)

Piggyback on responses to coordinating node

Test case	Throughput improvement %	50th % change	90th % change	99th % change
1 replica, no load	1.9%	-1.7%	0.5%	1.3%
1 replica, with load	115.8%	129.0%	-62.3%	-57.1%
4 replicas, no load	11.6%	-27.2%	-28.6%	-25.9%
4 replicas, with load	65.8%	-63.5%	-39.8%	-16.9%
1 replica, round robin, no load	6.8%	-7.2%	-16.6%	-28.0%



# Shrink & Split\*

5.0

6.1

---

\* Demo



# Shrink

Combine shards by a factor



# Split

Split into a factor of `number_of_routing_shards`

**Not required in 7.0+**

# Shards

7.0



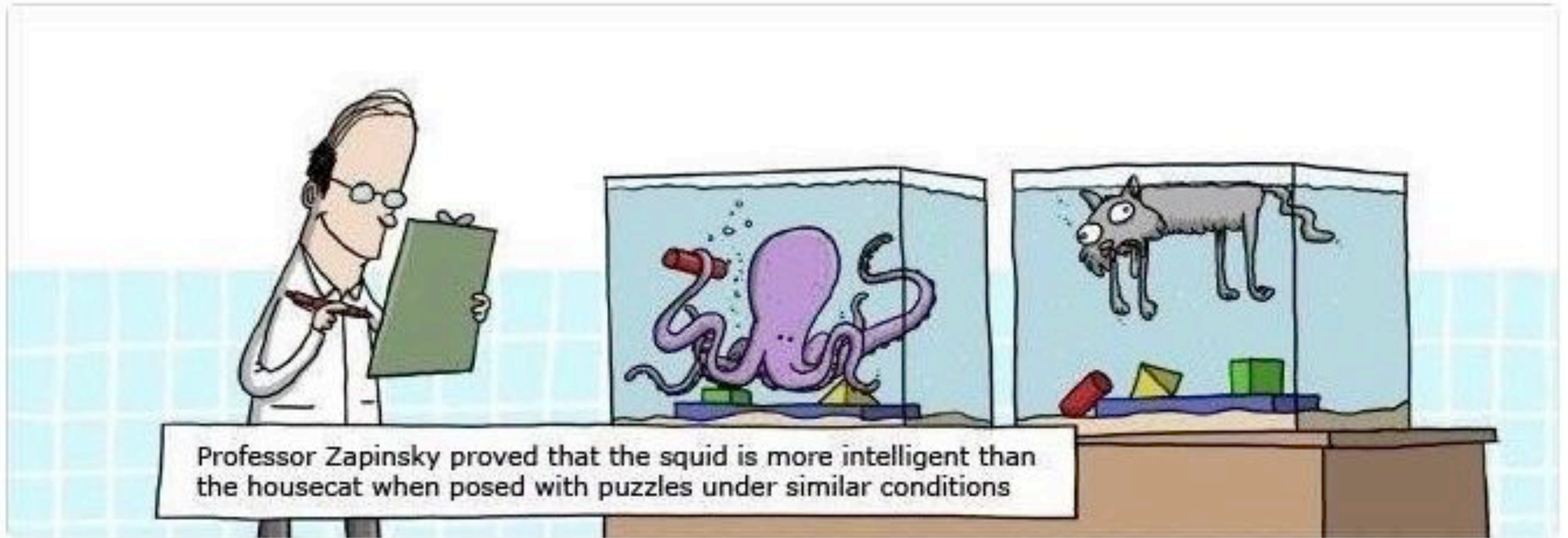
Default: 1 shard per index

Oversharding



# Benchmarks







# Rally

<https://elasticsearch-benchmarks.elastic.co>

# Conclusion

Strictness

Rolling Upgrades

Floodstage Watermark

Sequence Numbers

~~Types~~

Automatic Queue Resizing

Adaptive Replica Selection

Shrink & Split

Default Shard Number

meetup

<https://www.meetup.com/Greece-Elastic/>

Monday 19:00, The Cube Athens





# Questions?

Philipp Krenn

@xeraa