

# The Search for Speed

Scott Laird

Figma



Google

Google™  
Toolbar



WIDEVINE

YouTube

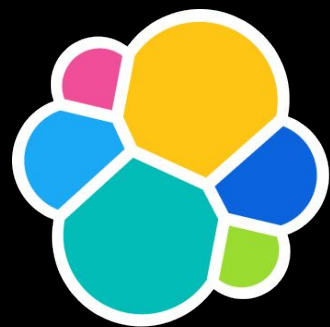


1. Latency is critical.
2. Relevancy is too.
3. Not serving errors is nice.
4. ???

1. Latency is critical.
2. Relevancy is too.
3. Not serving errors is nice.
4. *Profit?*

1. Latency is critical.
2. Relevancy is too.
3. Not serving errors is nice.
- ~~4. *Profit?*~~





elastic

Slow

Not resilient

Expensive

We didn't  
understand  
it.

The logo icon consists of a stylized blue and white shape resembling a drop or a leaf, with a curved line underneath it.

**OpenSearch**



**I HAVE NO  
IDEA WHAT  
I'M DOING**



No one understood  
what was happening  
under the hood.

Monitoring.  
(It's what we do.)

1. How many customer searches do we do per second?
2. How long do they take?
3. How many errors do we serve?

All of them  
were wrong.

**I HAVE NO  
IDEA WHAT  
I'M DOING**



So what *do*  
we know?

# Science

- Gather data
- Form a hypothesis
- Test the hypothesis
- Publish results
- Iterate

Gather data

Answers vary

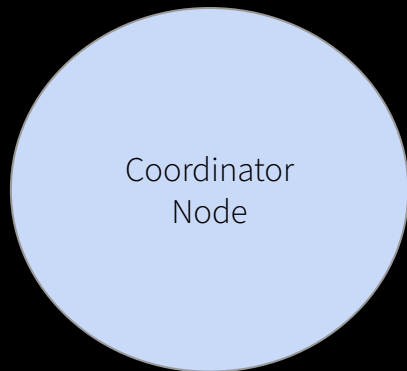
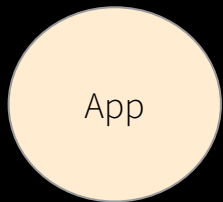
**I HAVE NO  
IDEA WHAT  
I'M DOING**



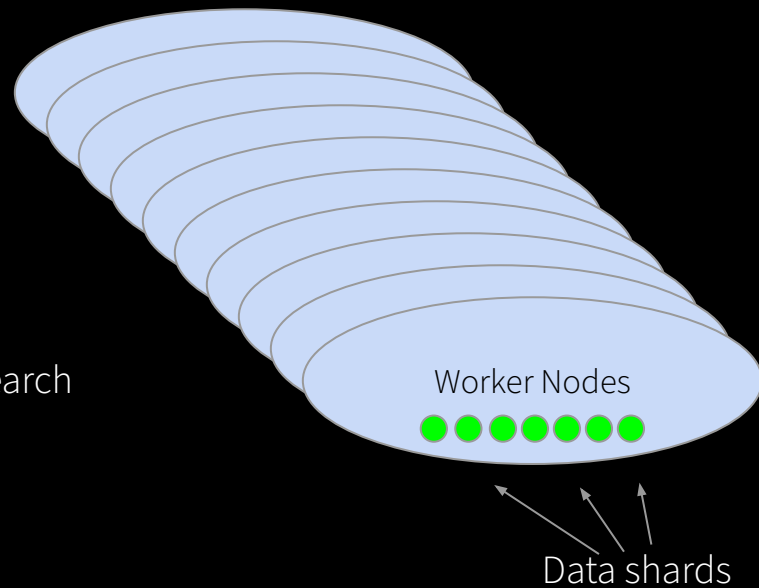
When in doubt,  
add more  
instrumentation

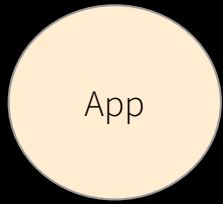


You need to know  
what you're measuring

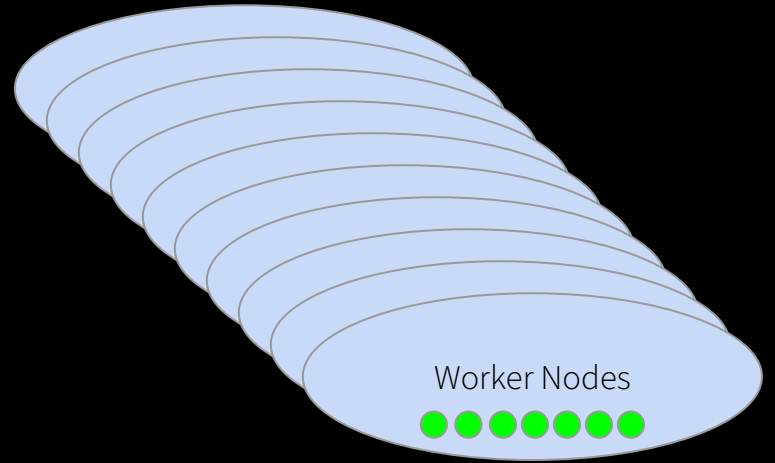
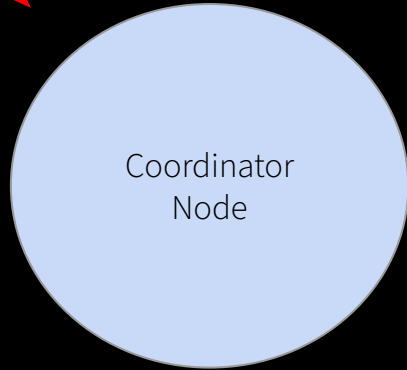


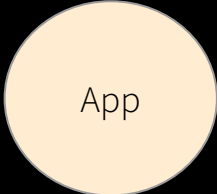
OpenSearch





Search for "foo"

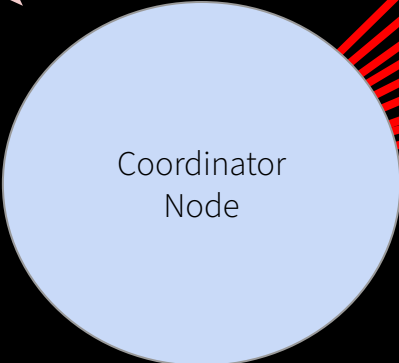




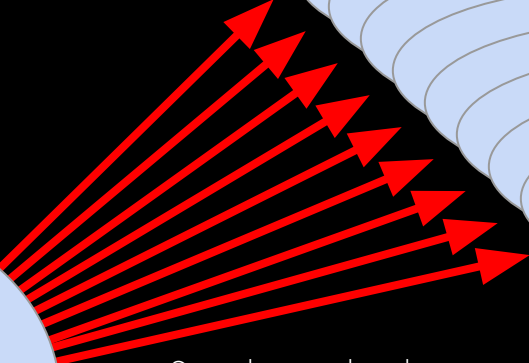
App



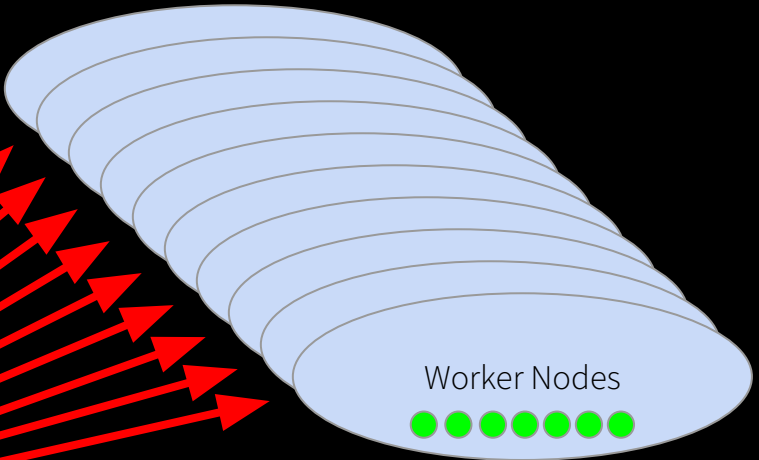
Search for "foo"



Coordinator  
Node

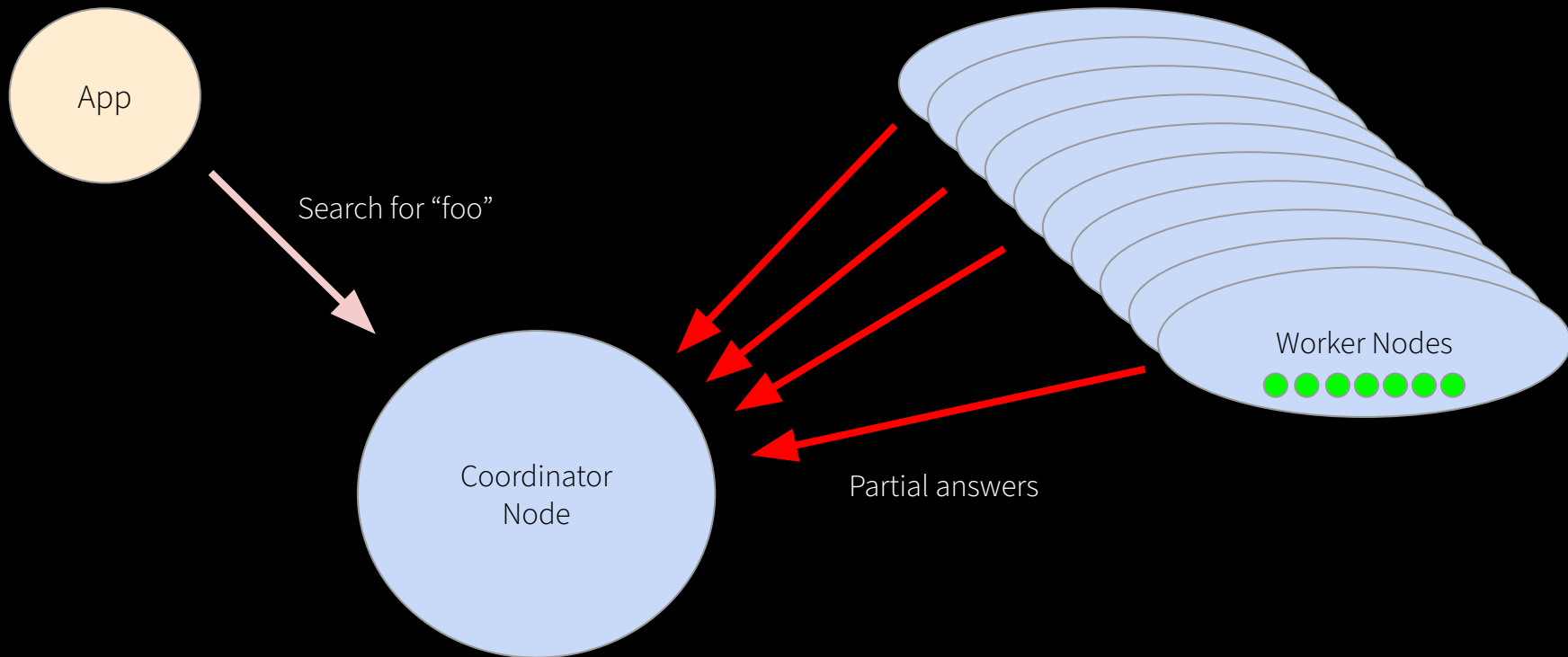


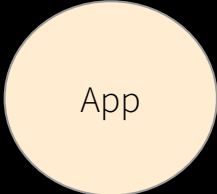
Search one shard  
for "foo"



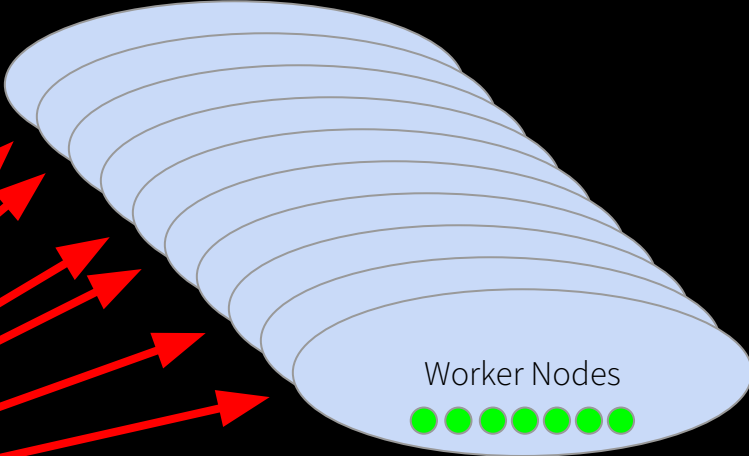
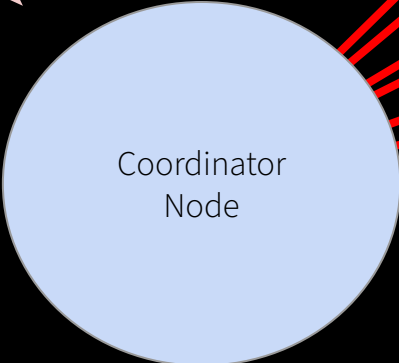
Worker Nodes



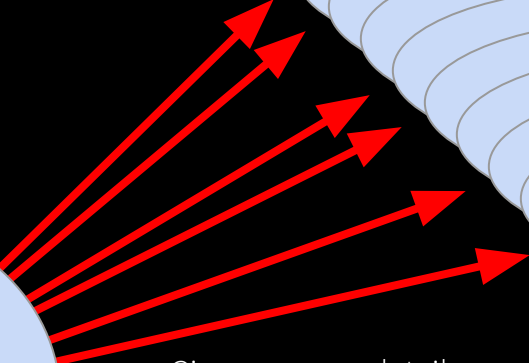


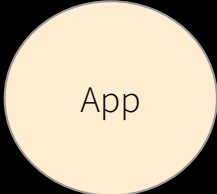


Search for "foo"

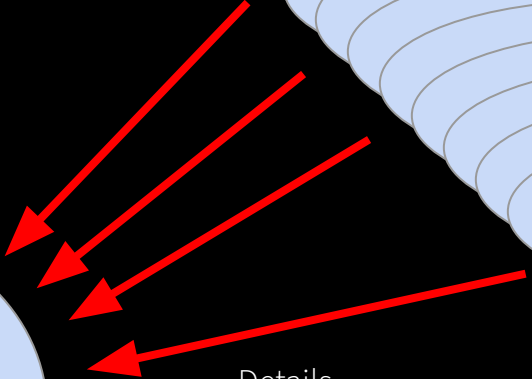
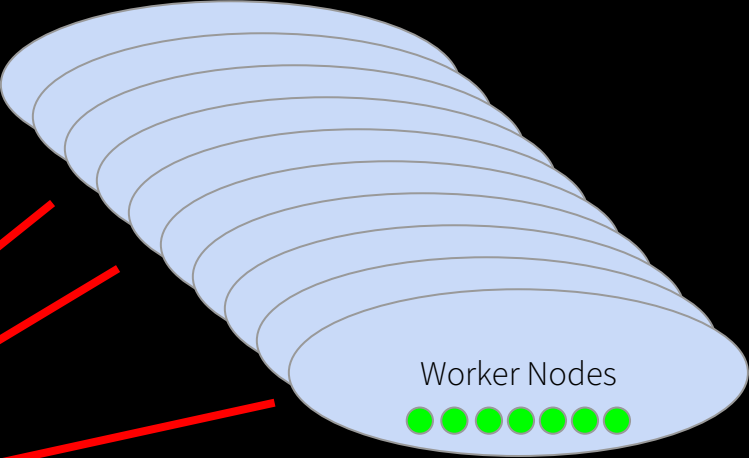
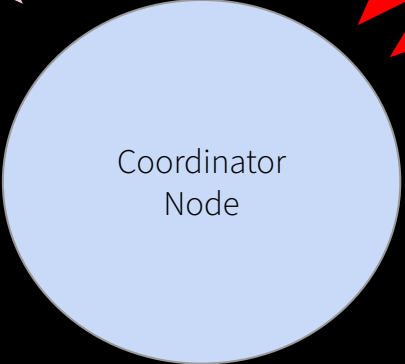


Give me more detail on document X

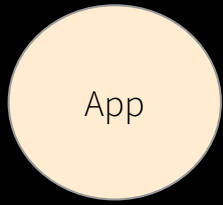




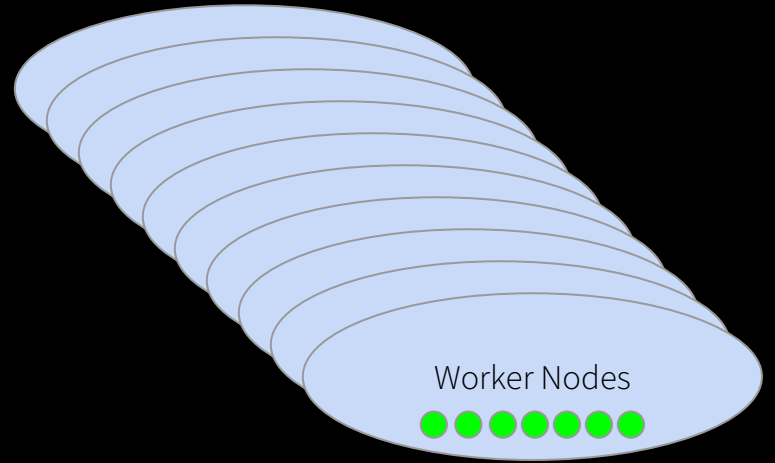
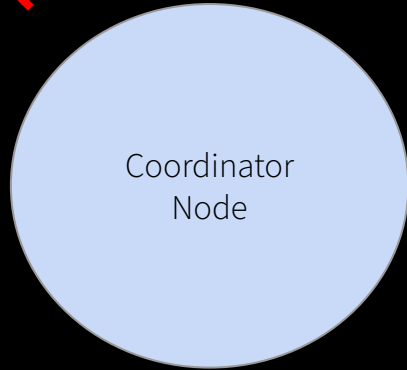
Search for "foo"



Details

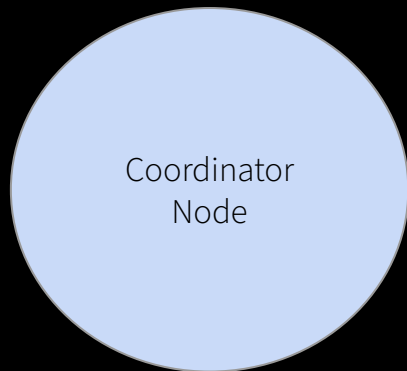
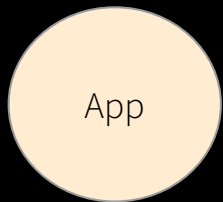


Results for "foo"

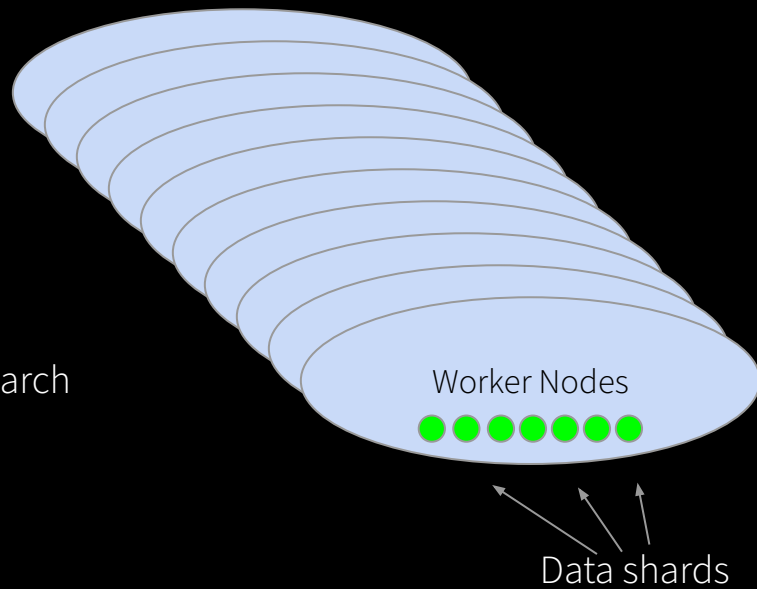


Open/Elastic Search  
metrics are per-shard

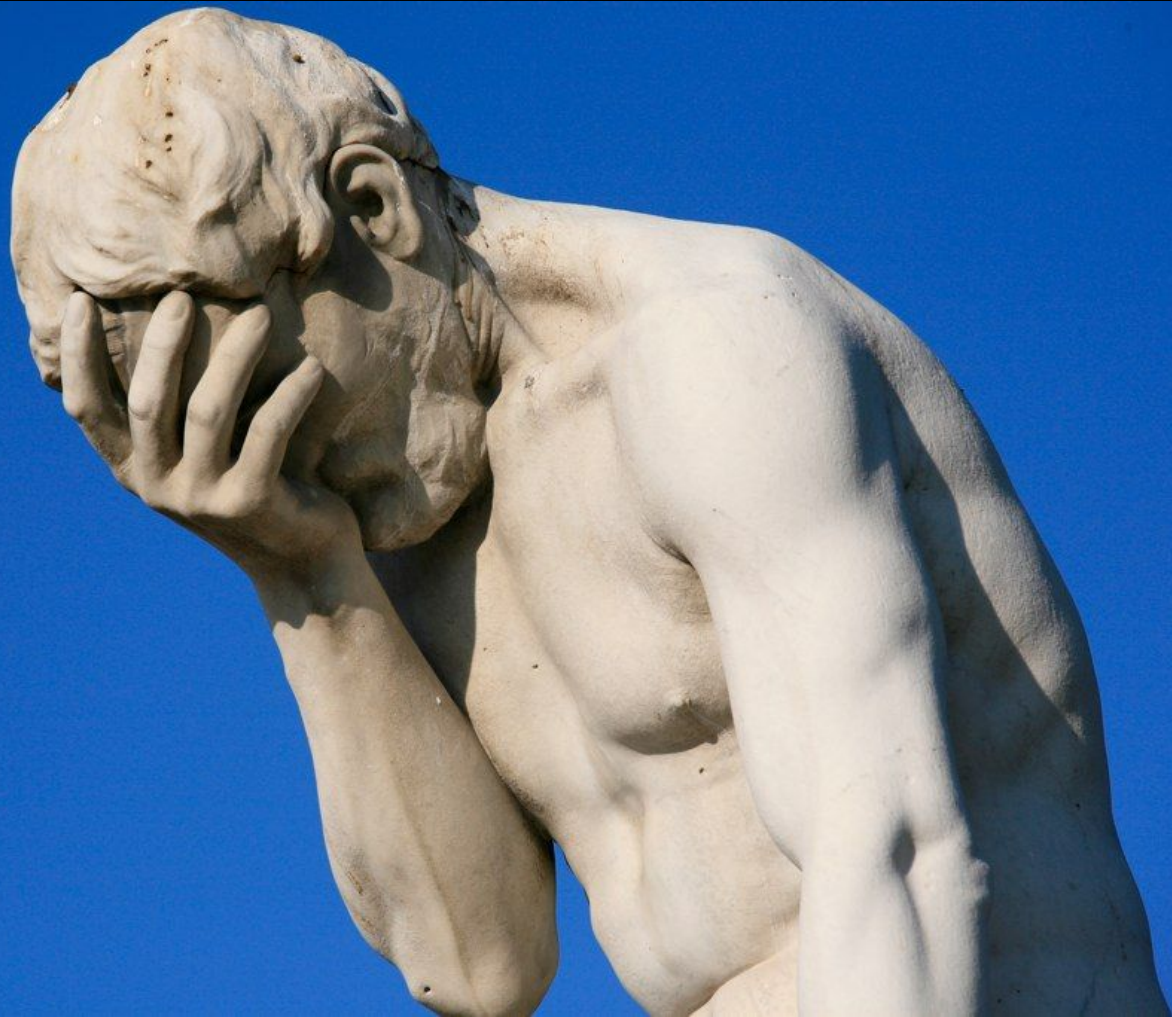
All of them



OpenSearch



Surely we're doing this  
wrong.



We're going to need to  
do this ourselves?

Slowly,  
things made sense.

Now we can fix things!

Time to start asking  
questions

A lot of the good questions  
were “if we changed X,  
what would happen?”

We're going to need a  
bigger boat.

We're going to need a  
bigger ~~boat~~.

We're going to need a  
bigger ~~boat~~ test cluster.

So how do we benchmark  
a bunch of test queries  
against OpenSearch?

opensearch-benchmark Public Watch 16 Fork 88 Star 116

main 16 Branches 23 Tags Go to file Add file Code

<b>gkamat</b>	Update OpenSearch version used for integ tests to 2.17 (#788)	3dc35c5 · 5 days ago	2,363 Commits
📁 .ci	update to JDK 21 for integ tests (#737)		2 months ago
📁 .github	Updated the Docker workflows to use the current versio...		3 weeks ago
📁 benchmarks	Provide fixes based on Issues #55 and #57 (#58)		4 years ago
📁 docker	More updates to the OSB build (#635)		6 months ago
📁 docs	Replaced occurrences of "Benchmark" with "OSB" (#705)		2 months ago
📁 it	Update OpenSearch version used for integ tests to 2.17 (...)		5 days ago
📁 jenkins	Address string issue in Groovy (#646)		6 months ago
📁 osbenchmark	Add @time_func annotation to sleep runner (#781)		2 weeks ago
📁 samples/ccr	Replaced occurrences of "Benchmark" with "OSB" (#705)		2 months ago
📁 scripts	Replaced occurrences of "Benchmark" with "OSB" (#705)		2 months ago
📁 tests	Add @time_func annotation to sleep runner (#781)		2 weeks ago
📄 .coveragerc	Add coverage configuration		9 years ago

About

OpenSearch Benchmark - a community driven, open source project to run performance tests for OpenSearch

[opensearch.org/docs/latest/benchmark...](https://opensearch.org/docs/latest/benchmark...)

- 📖 Readme
- 📄 Apache-2.0 license
- 📄 Code of conduct
- 📄 Security policy
- 📄 Activity

- 📄 Custom properties
- ☆ 116 stars
- 👁 16 watching
- 🍴 88 forks

Report repository

Releases 23

1.12.0 Latest 3 weeks ago

+ 22 releases

or maybe not...

benchmark ALL THE  
THINGS!



So we measured staging...

So we measured staging++...

So we measured staging+++...

Then we measured prod.

Then we measured a clone of prod.

We changed config flags

We changed the number of shards

We changed AWS instance types

We tested different versions

None of the things we'd  
expected were true

(Almost) None of the things we'd  
expected were true

To go faster, we needed  
fewer shards

Adding nodes didn't help

Most of the go-fast config options  
slowed us down

Somehow, our queries  
were nearly optimal

Upgrading OpenSearch mostly  
didn't help or hurt

Swapping for nodes with less CPU (but more RAM) was a big win

So we rolled that all out

That was a good start

But wait, there's more

When all you have is a giant pile of trace data, everything looks like code that can be optimized

Less data is best data

Faster is better

What Didn't Work

Painfully Generic Pain Points  
(obligatory SRE rant)

# Concrete Recommendations

# For {Open,Elastic}Search

- Keep a close eye on IOPS
- Keep indexes as small as possible
- Keep shard count small
- Keep shards evenly distributed across nodes

# For everything

- Run experiments.
- Measure things.
- Write up and share your results.
- SLOs are *magic*.

Q and/or A?