DataStax

Developers

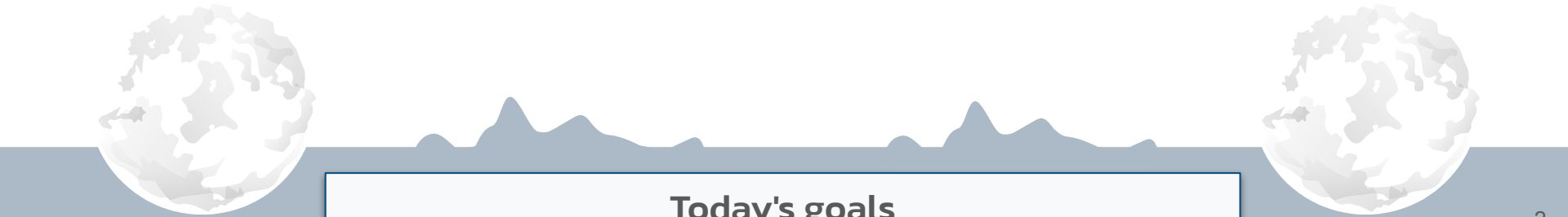
Benchmark your database with NoSQLBench

A serious performance testing tool for NoSQL data systems

Learn about testing a distributed NoSQL database (in the right way, at the right time)

Play with the powerful benchmarking tool **NoSQLBench** on a real DB

Try out the basics, peek at more advanced features, know where to go for more!



01

Housekeeping

Live and Hands-on

02

Benchmarking

Why, what, how?

03

NoSQL Database

Cassandra & Astra DB

04

NoSQLBench

A tool with superpowers

05

Practice

Hands-on benchmarking!

06

What's next?

Quiz, Homework, Resources

Housekeeping

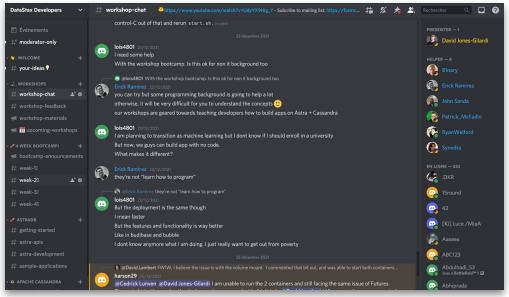
Live and Hands-on

Livestream: youtube.com/DataStaxDevs

YouTube

Twitch

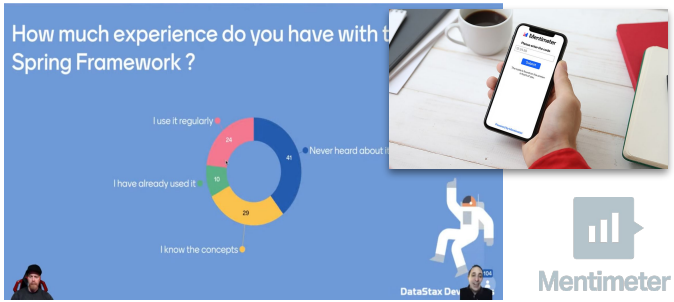
Questions: https://dtsx.io/discord



YouTube

Discord

Games and quizzes: menti.com

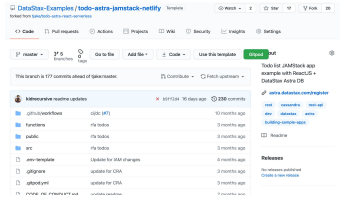


Mentimeter

Live Sessions

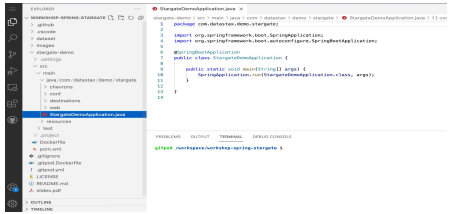
Nothing to install !

Commands + exercises + slides



GitHub

IDE



Gitpod

Distributed NoSQL Database

DataStax
Astra DB

Hands-On Housekeeping

01

Housekeeping

Live and Hands-on

02

Benchmarking

Why, what, how?

03

NoSQL Database

Cassandra & Astra DB

04

NoSQLBench

A tool with superpowers

05

Practice

Hands-on benchmarking!

06

What's next?

Quiz, Homework, Resources

Benchmarking

Why, what, how?

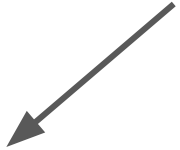
"Will the data system behind my app work as I want it to?"

"Will my data model work correctly?"

functional testing

"Will it support my millions of users?"
"Will it respond fast enough?"
"Will it transfer data at a sufficient rate?"

operational testing (performance)



Confidence/knowledge of your SLO (*service level objectives*)
Convincing decision makers about adopting a technology
Part of the assessments before a migration

Why "benchmarking" ?

Focus on operational testing (performance). Well-defined goals:

"Will it support my millions of users?" → testing scalability

"Will it respond fast enough?" → testing response time ("latency" *)

"Will it transfer data at a sufficient rate?" → testing throughput

We want to test **realistic workloads**, similar to what the actual application will do

Assumption: a NoSQL distributed database being used

Performance testing, strategies

"Distributed databases are hard to observe/measure"

- Must emulate realistic workload.
- High variance in responses under the best of conditions ("flakiness").

What can one do?

- Server-side instrumentation: powerful, but costly (and: not always access to DB internals)
- End-to-end ("black-box"): a.k.a. instrument the client
  - Will use public API ("no cheating").
  - Don't overdo (cost, maintenance).
  - Work with predictable environments (no snowflakes).

"A client sending reads/writes to the database and timing everything"

How ?

01

Housekeeping

Live and Hands-on

02

Benchmarking

Why, what, how?

03

NoSQL Database

Cassandra & Astra DB

04

NoSQLBench

A tool with superpowers

05

Practice

Hands-on benchmarking!

06

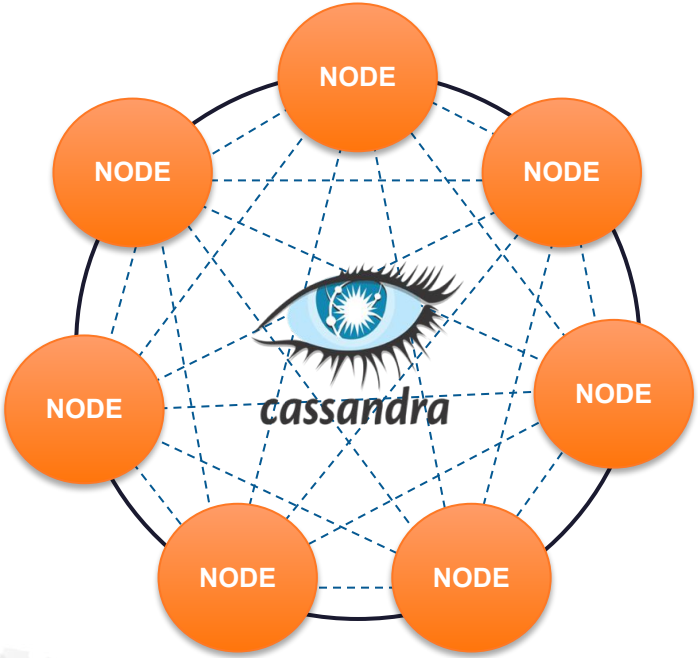
What's next?

Quiz, Homework, Resources

NoSQL Database

Cassandra & Astra DB

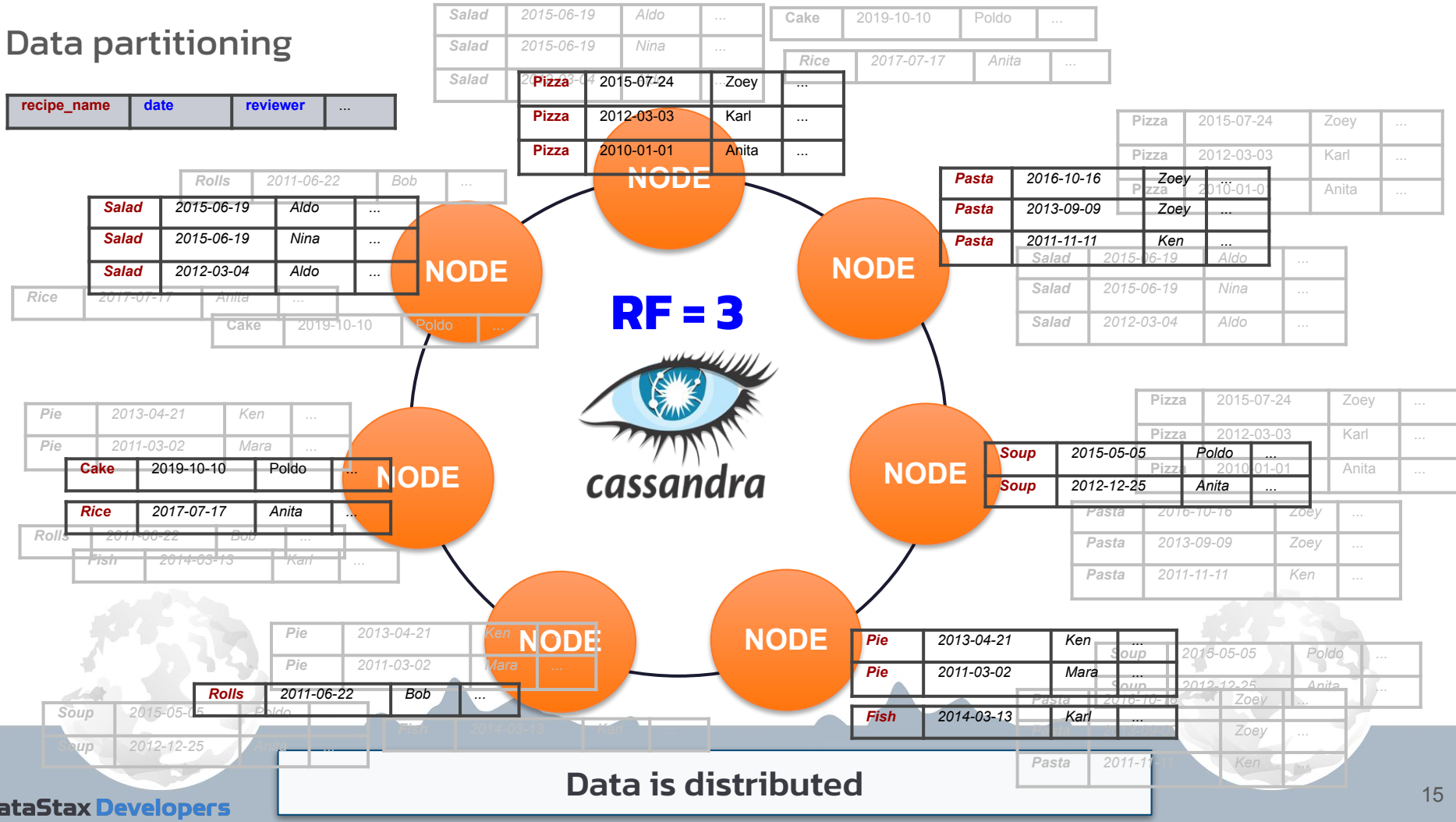
Apache Cassandra™: a powerful NoSQL database



- Big Data Ready
- Read / Write Performance
- Linear Scalability
- Highest Availability
- Self-Healing and Automation
- Geographical Distribution
- Platform Agnostic
- Vendor Independent

Data partitioning

recipe_name	date	reviewer	...



Pizza	2015-07-24	Zoey	...
Pizza	2012-03-03	Karl	...
Pizza	2010-01-01	Anita	...

Salad	2015-06-19	Aldo	...
Salad	2015-06-19	Nina	...
Salad	2012-03-04	Aldo	...

Pasta	2016-10-16	Zoey	...
Pasta	2013-09-09	Zoey	...
Pasta	2011-11-11	Ken	...

Soup	2015-05-05	Poldo	...
Soup	2012-12-25	Anita	...

Pie	2013-04-21	Ken	...
Pie	2011-03-02	Mara	...

Cake	2019-10-10	Poldo	...
Rice	2017-07-17	Anita	...

Rolls	2011-06-22	Bob	...
Fish	2014-03-13	Karl	...

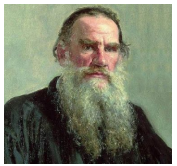
Data is distributed

Avoid "whole-table" queries!

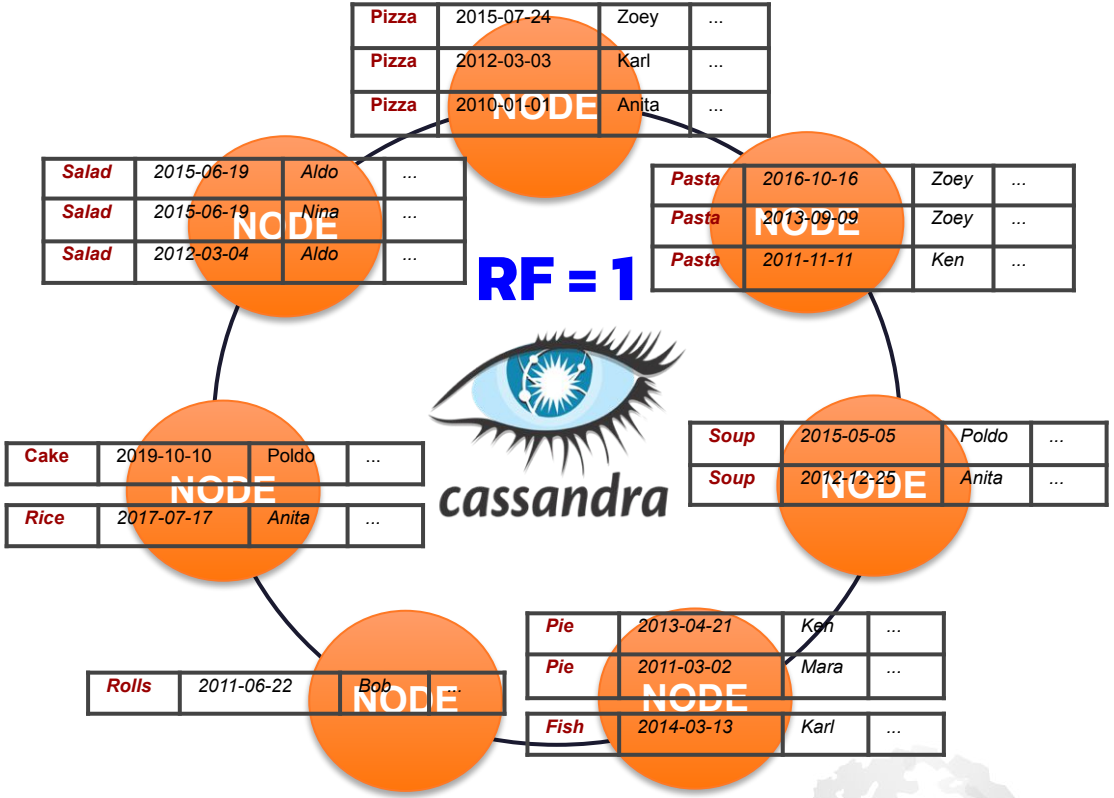
Carefully plan tables **after** thinking of the application

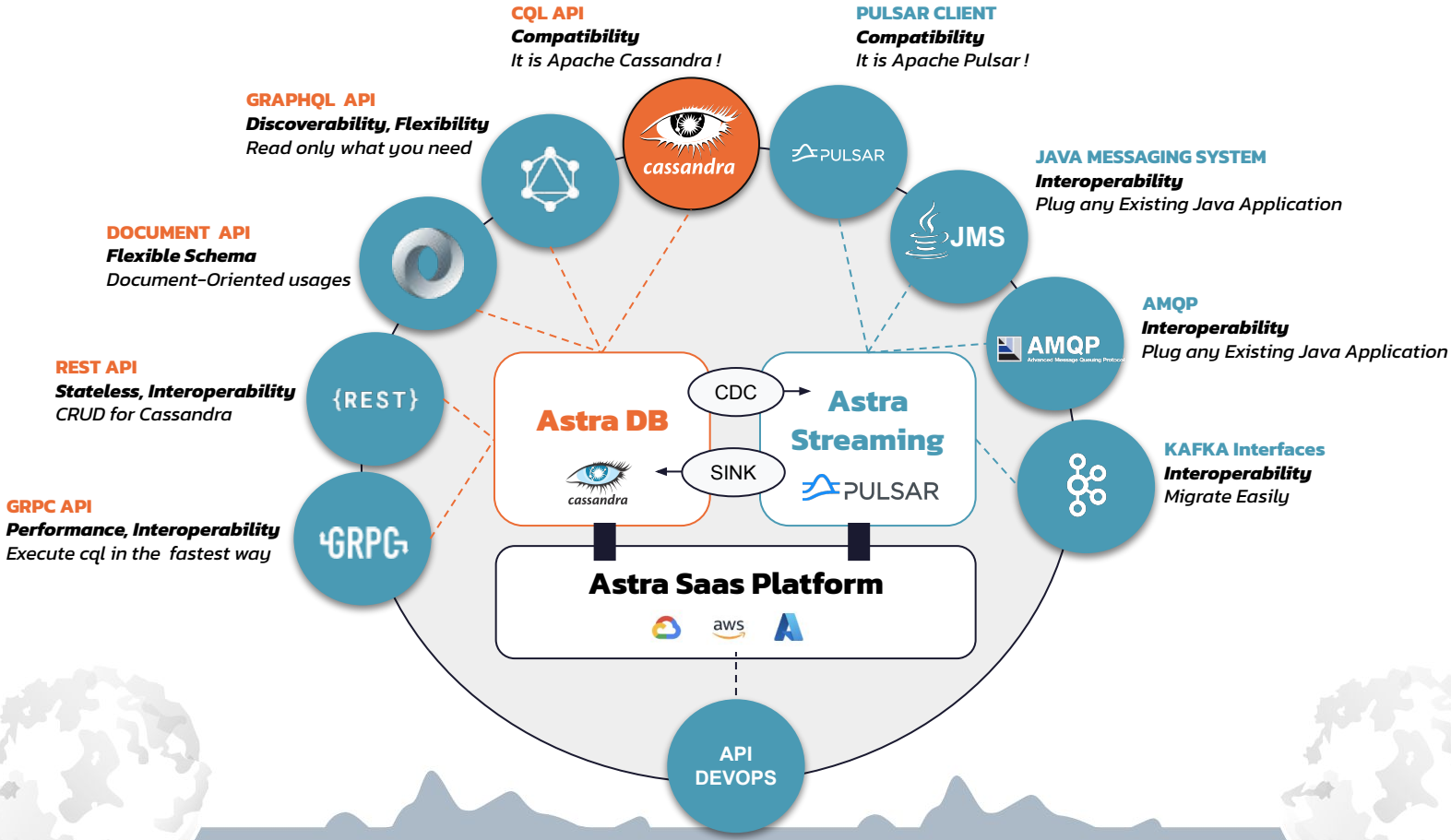
Denormalization is your friend

No JOINs!

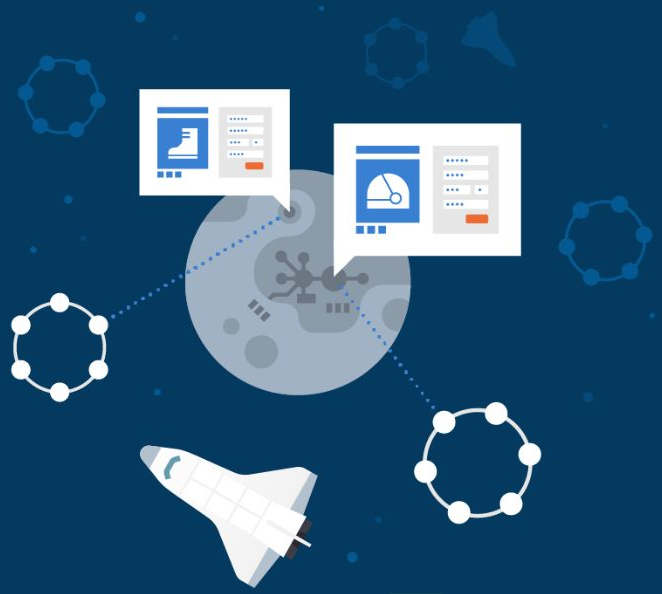


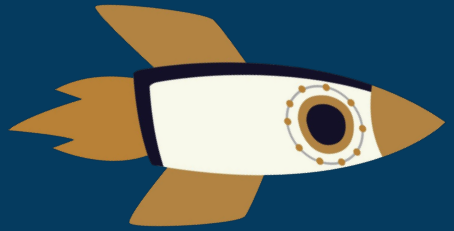
"All relational databases are alike; each NoSQL database is NoSQL in its own way"





Astra DB = managed Cassandra (*and much more*)



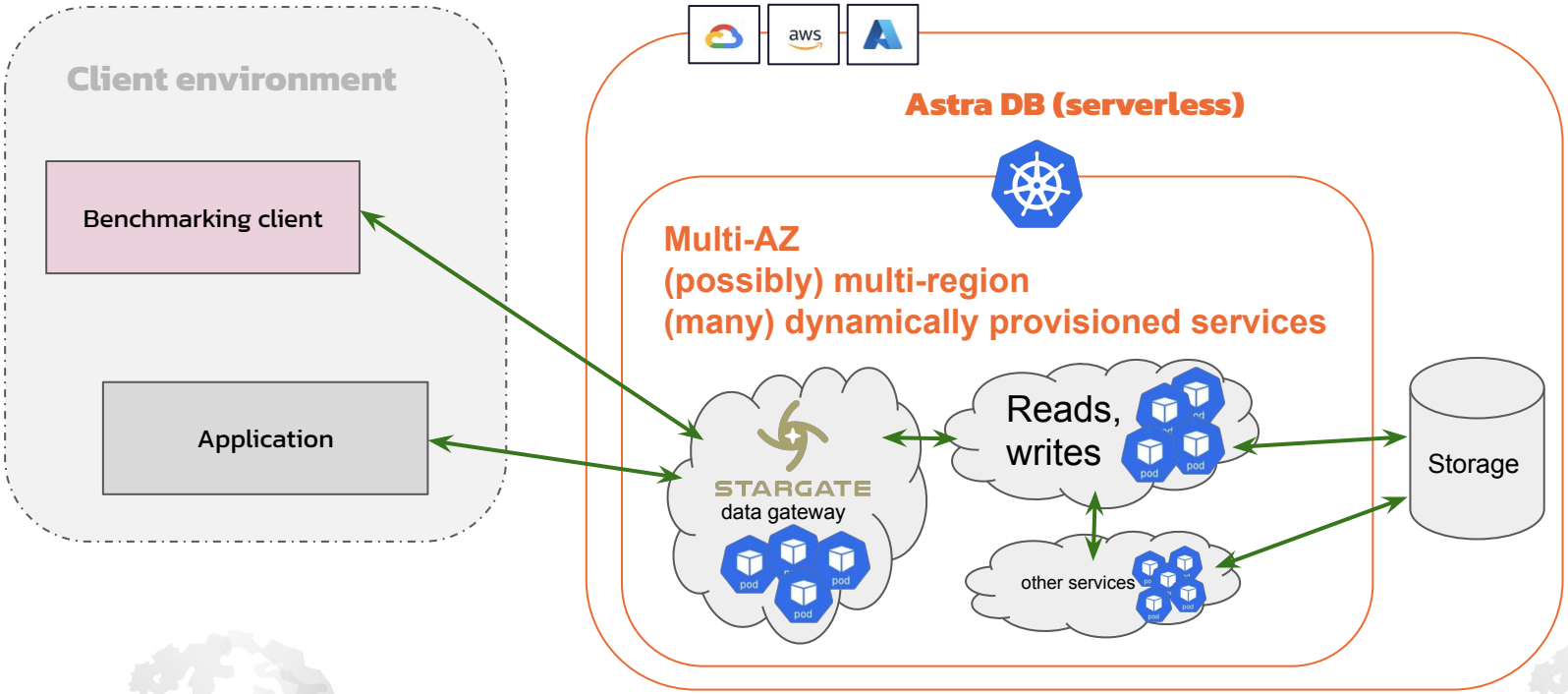


Hands-on (!github)

Database setup

- ✅ Create your database
- ✅ Create a token
- ✅ Download secure-connect-bundle

Data I/O with Astra DB



Astra DB: architecture sketch

Physical distances (& networking) matter

Benchmarking client

Where is the testing client located?

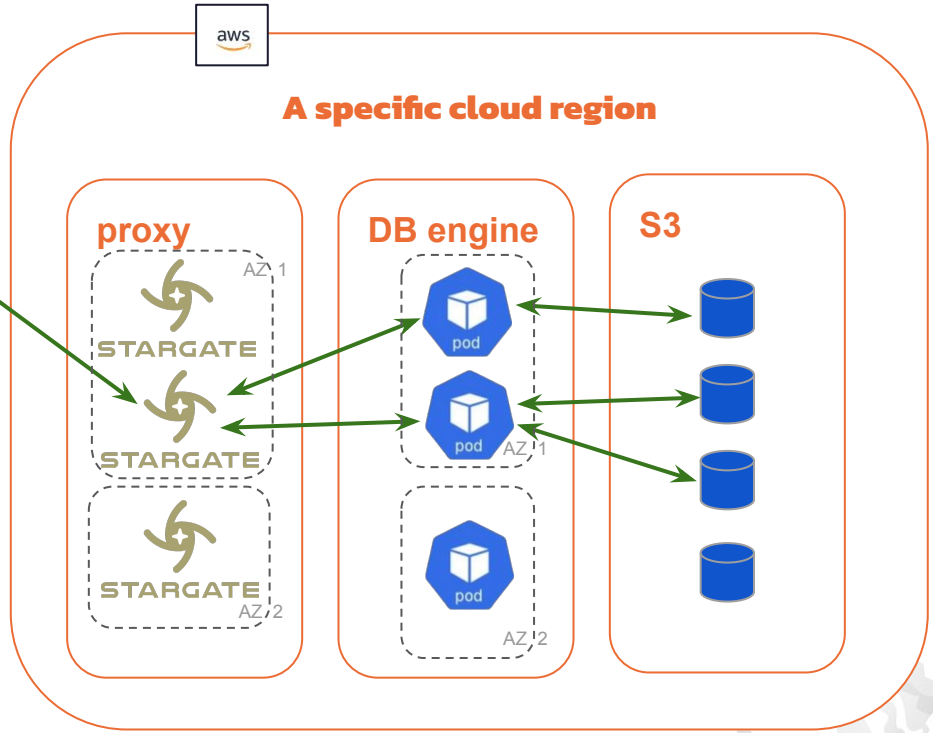
- Is it in AWS?
- Is it in the same region?

Where *should* it be?

- *"workload just like the actual app..."*

Cloud makes it all ambiguous:
what "latency" shall I measure?

Benchmarking on cloud: still relevant



An example request

01

Housekeeping

Live and Hands-on

02

Benchmarking

Why, what, how?

03

NoSQL Database

Cassandra & Astra DB

04

NoSQLBench

A tool with superpowers

05

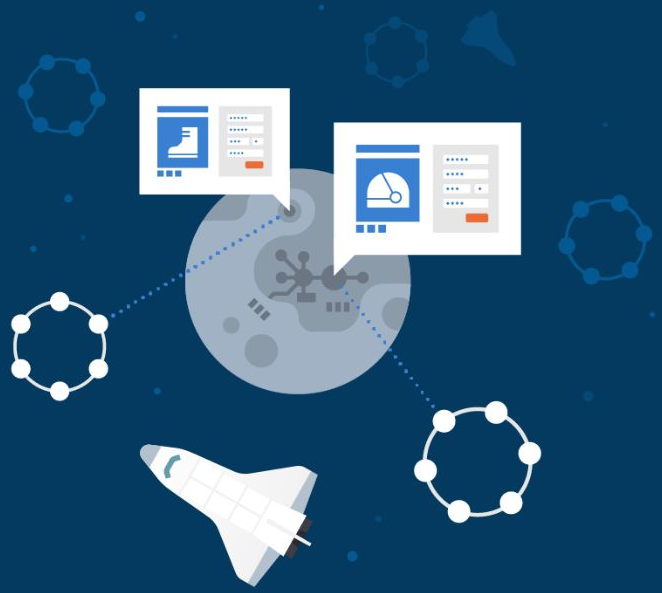
Practice

Hands-on benchmarking!

06

What's next?

Quiz, Homework, Resources



NoSQLBench

A tool with superpowers

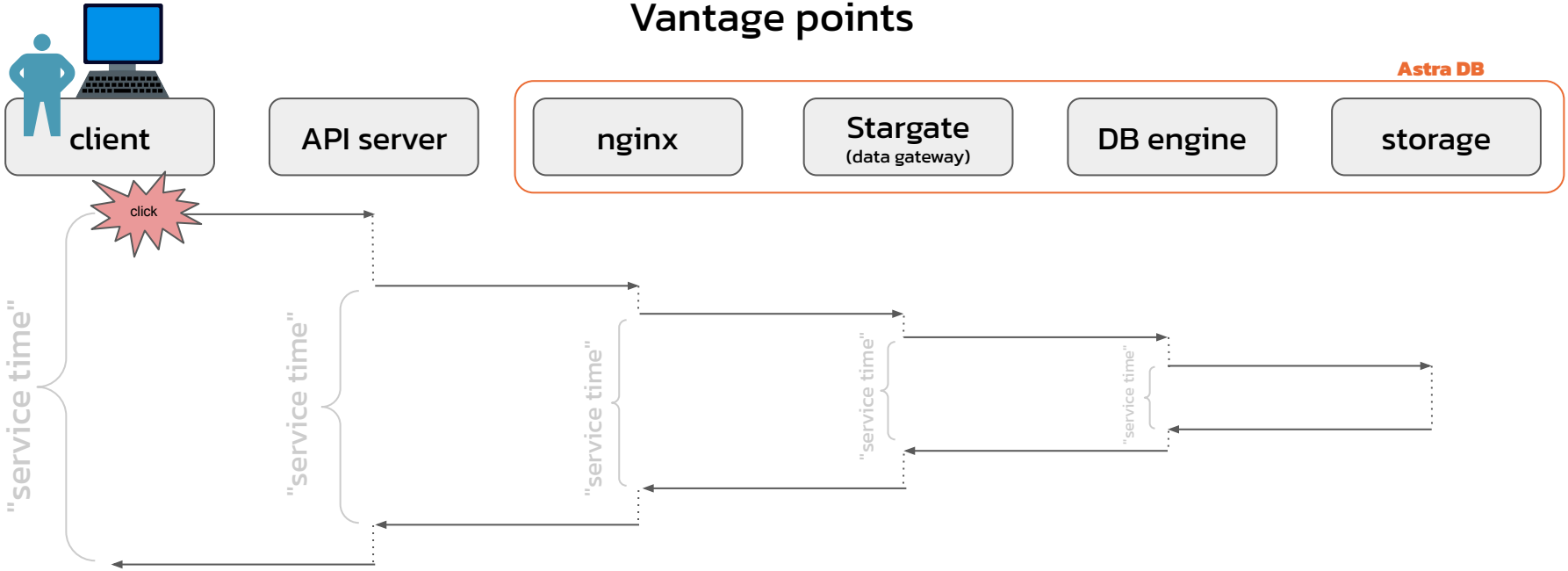
(e2e) performance benchmarking in practice

"A client sending reads/writes to the database and timing everything"

But:

- **realistic synthetic testing**:
  - workload "just like the actual app"...
  - ... but before the app even exists
- reproducible workloads to the details (determinism)
- don't want the client overhead to get in the way of measurements (... not too much)
- distributed system, cloud: where to run the testing client? (just *what* should be tested?)
- the ideal tool: highly customizable but also some off-the-shelf tests (i.e. ready-to-use as they are)

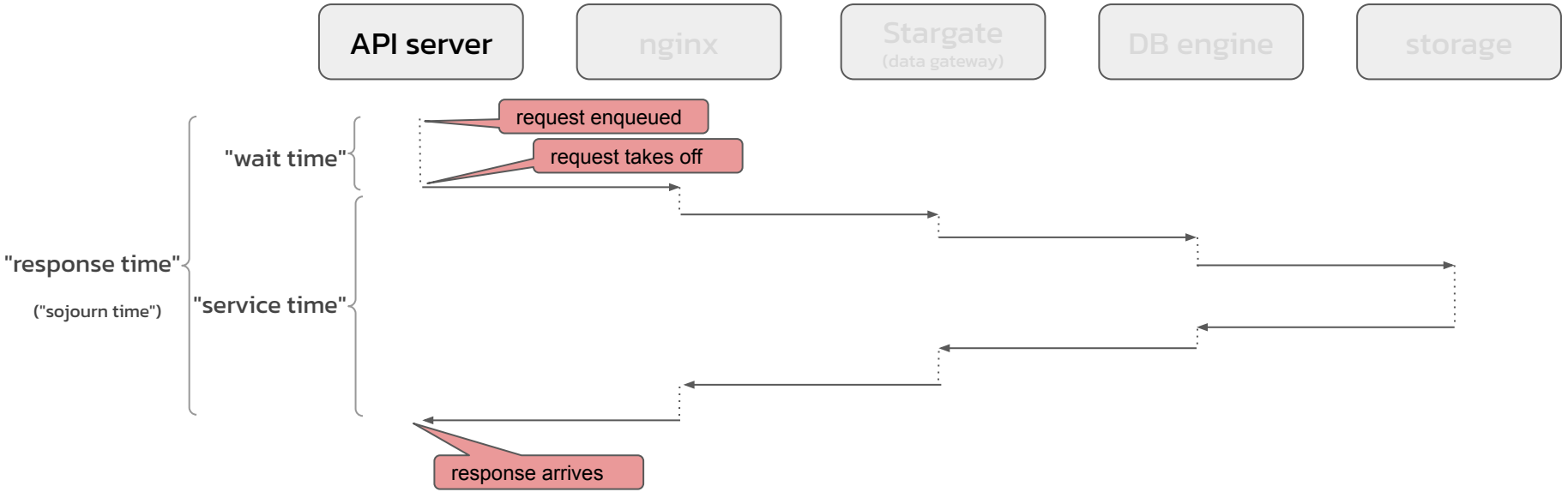
Vantage points



Focus on the "outermost layer" first (users!)

"Latency", for whom?

As seen by the API:

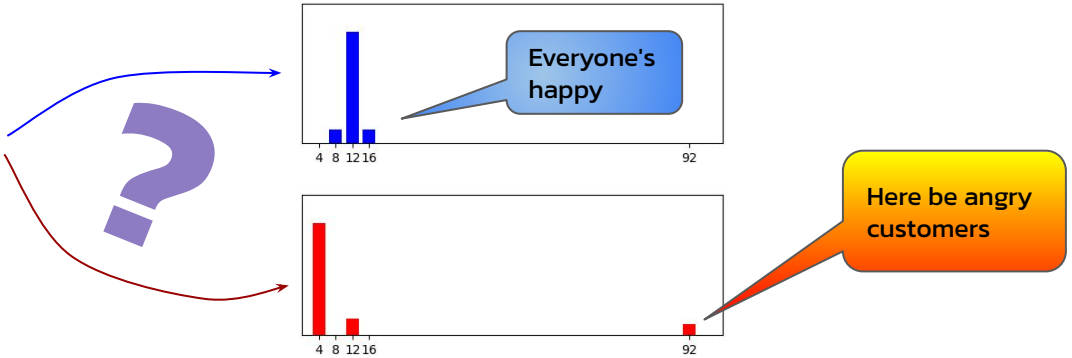


Risk: "coordinated omission" (benchmark bias from congested queue)

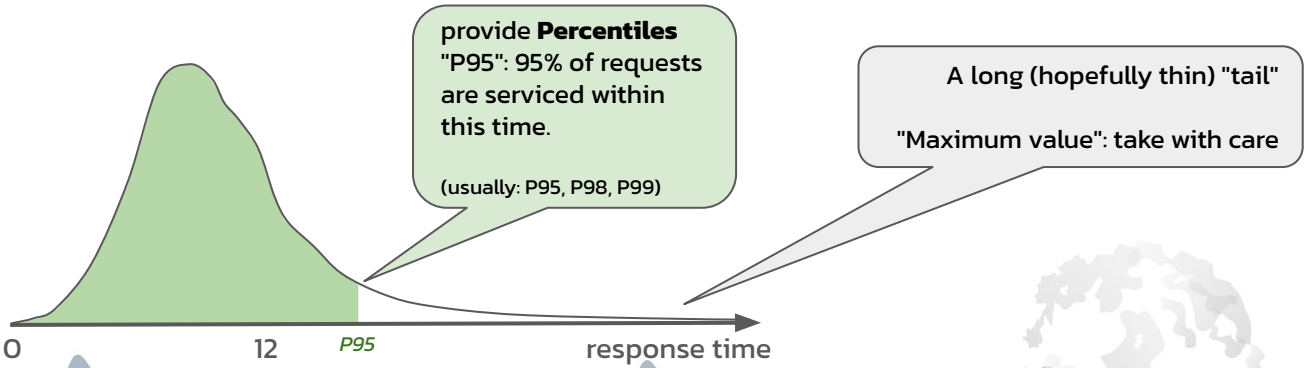
Subtleties with service time

Characterizing performance

"Average response time = 12 ms"



Better: (probability) distributions



Distributions

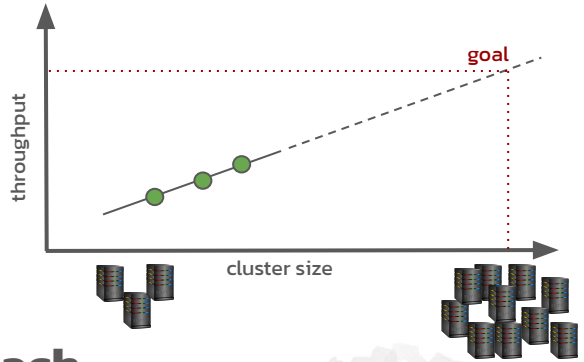
Example approach

"How large the cluster to achieve given **response time** and **throughput** ?"

First: "scale-up" estimate to check response time

Then: "scale-out" analysis + extrapolation

(no need to burn $ with actual scales, just make sure you're in the "scaling regime")



"Vertical-then-horizontal" approach

Overall methodology

Curves & Little's law

$\lambda = L / W$

Average arrival rate in supermarket = (average number of people in it) / (average time spent shopping)

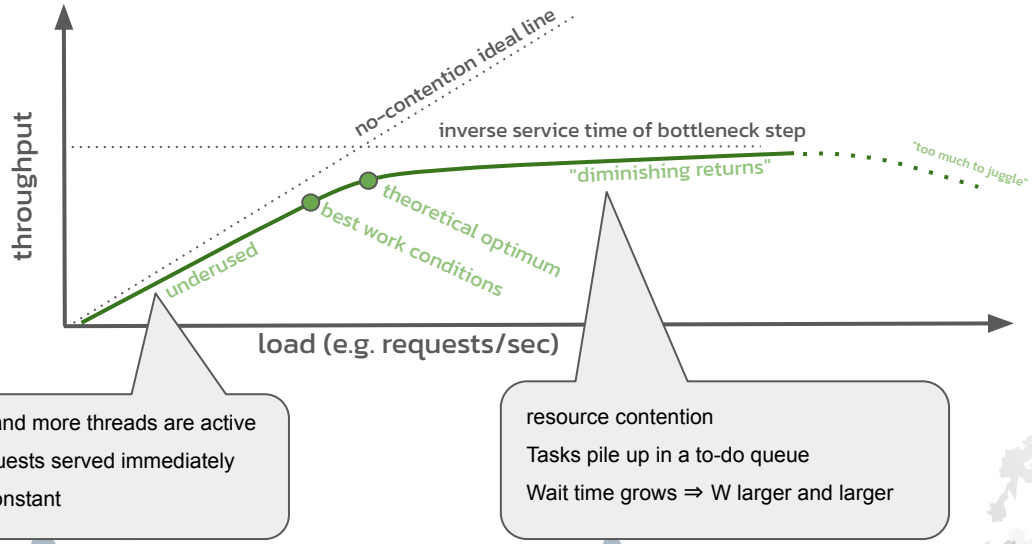
$L = \lambda * W$

Average # requests waiting or being executed = (rate requested for benchmark) * (service time)

$W = L / \lambda$

Average time bottles spend in cellar = (average count of bottles) / (frequency of picking a bottle)

Systematic load testing



more and more threads are active

all requests served immediately

W ~ constant

resource contention

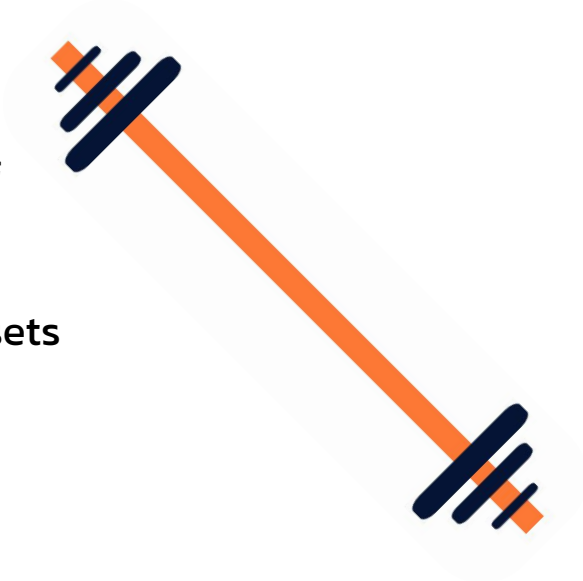
Tasks pile up in a to-do queue

Wait time grows ⇒ W larger and larger

The limits of a system

Some key features

- keeps "coordinated omission" in check
- minimizes impact of measurement infrastructure itself
- metrics to verify client is not overloaded
- Virtual Data Sets, for lightweight, replayable, big data sets
- drivers for many DBs; extensible architecture
- many ways to get and inspect results
- ...



Today's benchmarking tool

01

Housekeeping

Live and Hands-on

02

Benchmarking

Why, what, how?

03

NoSQL Database

Cassandra & Astra DB

04

NoSQLBench

A tool with superpowers

05

Practice

Hands-on benchmarking!

06

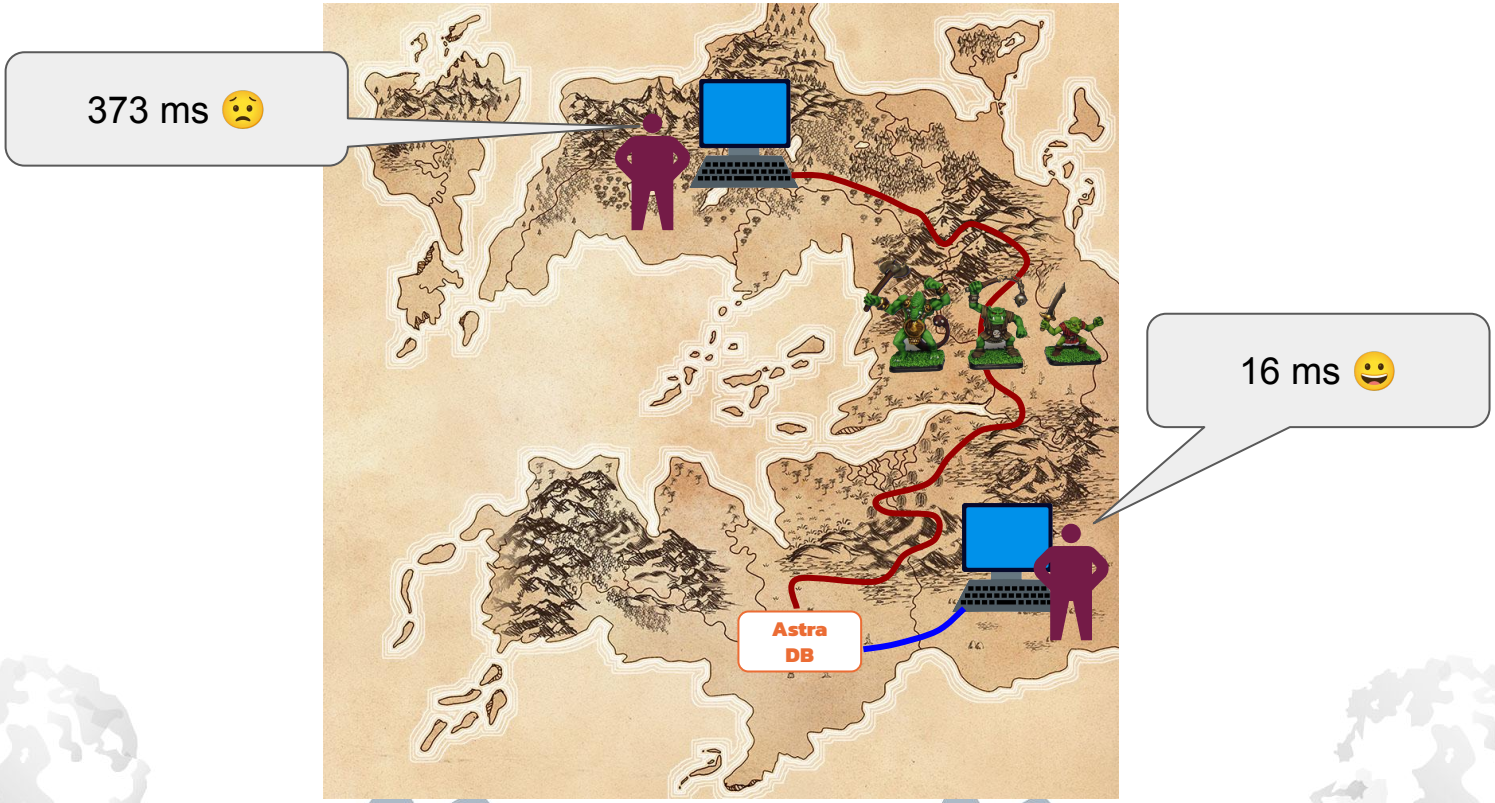
What's next?

Quiz, Homework, Resources

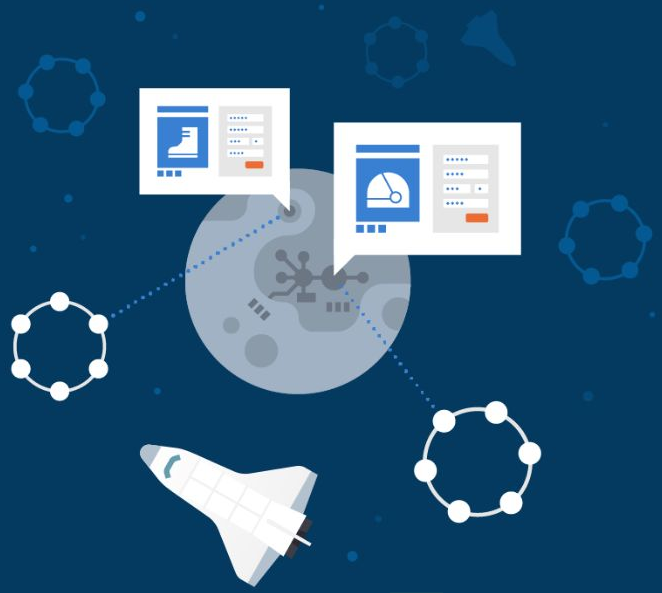
Practice

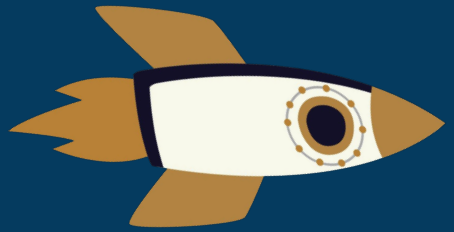
Hands-on benchmarking!

Today: client machines kindly provided by Gitpod. *Where are they?*



A word of caution





Hands-on (!github)

Use NoSQLBench

- ✅ Install & start benchmarks
- ✅ The many ways to inspect results
- ✅ Build custom workloads

01

Housekeeping

Live and Hands-on

02

Benchmarking

Why, what, how?

03

NoSQL Database

Cassandra & Astra DB

04

NoSQLBench

A tool with superpowers

05

Practice

Hands-on benchmarking!

06

What's next?

Quiz, Homework, Resources

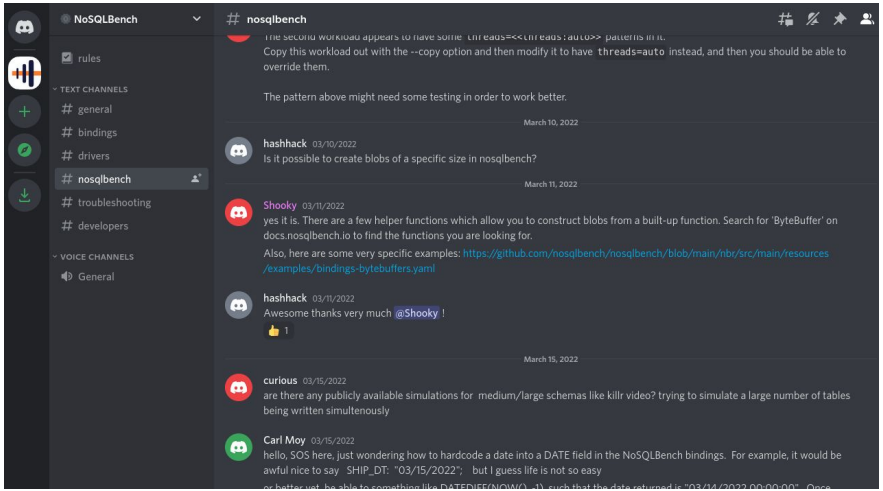
What's next?

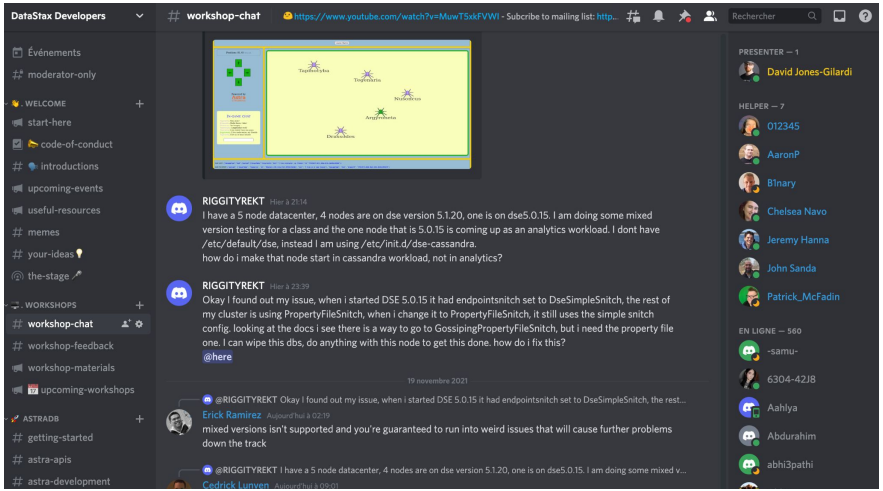
Quiz, Homework, Resources

discord.gg/dBHRakusMN

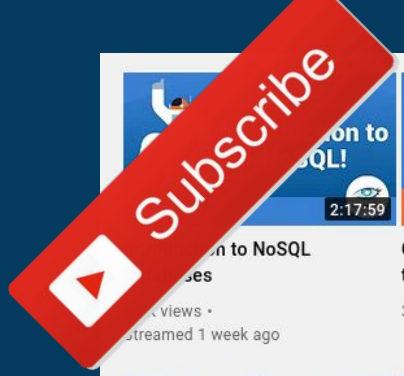
dtsx.io/discord

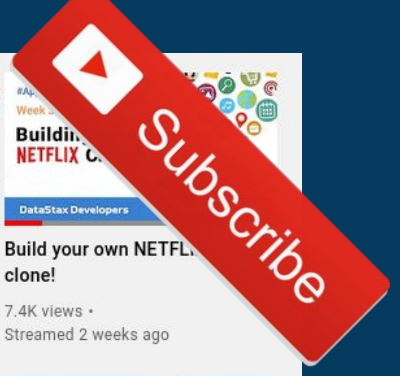
DataStax





Discord, Discord, Discord





…n to NoSQL …ses

…k views •
…Streamed 1 week ago

Crash Course | Introduction to Cassandra for Developers

331 views • 1 week ago

Introduction to NoSQL Databases

3.5K views •
Streamed 1 week ago

Introduction to NoSQL Databases

10K views •
Streamed 1 week ago

Build your own NETFLIX clone!

4K views •
Streamed 2 weeks ago

Build your own NETFL… clone!

7.4K views •
Streamed 2 weeks ago

Astra Streaming Demo

177 views • 2 weeks ago

Kubernetes Ingress Management with Traefik…

496 views •
Streamed 2 weeks ago

Build your own TikTok clone!

1.9K views •
Streamed 3 weeks ago

Build your own TikTok Clone!

4K views •
Streamed 3 weeks ago

How to use the Connect Driver in Astra DB

113 views • 4 weeks ago

How to use the CQL Console in Astra DB

39 views • 4 weeks ago

How to create an Authentication Token in…

37 views • 4 weeks ago

How to use the Data Loader in Astra DB

62 views • 4 weeks ago

Astra DB Sample App Gallery

36 views • 4 weeks ago

How to use Secure Connect in Astra DB

42 views • 4 weeks ago

Cassandra Day India: CL Room (Workshops)

2.4K views •
Streamed 4 weeks ago

Cassandra Day India: RF Room (Talks)

1.3K views •
Streamed 1 month ago


INTRODUCTION TO
cassandra
WORKSHOP
DataStax Developers


cassandra
Introduction to
NoSQL
WORKSHOP
DataStax Developers


cassandra
DATA MODELING
WORKSHOP
DataStax Developers


cassandra
SQL to NoSQL
WORKSHOP
DataStax Developers


K8SSANDRA
WORKSHOP
DataStax Developers


cassandra
Spring Data +
Cassandra
WORKSHOP
DataStax Developers


cassandra
Quarkus +
Cassandra
WORKSHOP
DataStax Developers


Vue.JS
WORKSHOP
DataStax Developers


ReactNative
WORKSHOP
DataStax Developers


GraphQL
WORKSHOP
DataStax Developers


Building a
ToDO App
WORKSHOP
DataStax Developers


TikTok Clone
WORKSHOP
DataStax Developers


Netflix Clone
WORKSHOP
DataStax Developers


Event Streaming
Learning Series
DataStax Developers


Event
Streaming
WORKSHOP
DataStax Developers


Reactive
Spring
WORKSHOP
DataStax Developers


AI as an
API
WORKSHOP
DataStax Developers


Change
Data Capture
WORKSHOP
DataStax Developers


Build an
e-commerce a
WORKSHOP
DataStax Developers


NoSQL
Bench
WORKSHOP
DataStax Developers

Badges

Thank You!

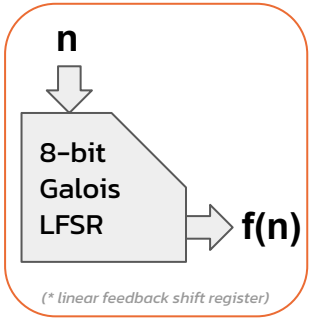
Generating data for benchmarks

We want

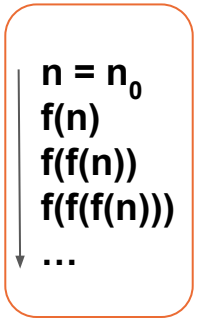
- a huge amount of random-looking data (but: reproducible)
- … without having to store it all, juggle it to/from disk, etc
- … with the ability to retrieve the *n*th value in the sequence in constant time for any *n*

Virtual Data Sets to the rescue!

With this:



build this:



by tuning the LSFR, the sequence spans all the 2^n-1 values in *one cycle* (whatever the nonzero n_0).

Then we can turn this cycle on its head and define:

n*th value in sequence = *f(n)