

YCSB

**Brian F. Cooper, Adam Silberstein, Erwin Tam,
Raghu Ramakrishnan, Russell Sears**

Brian's guide to writing a widely cited paper

1. Work in a new-ish, hot area
2. Discover that there is no good way to compare different systems
3. Come up with something halfway reasonable
4. Make it super easy for people to use

Funny story #1

So Raghu was invited to give a keynote at VLDB...

Funny story #1

So Raghu was invited to give a keynote at VLDB...



Hey Brian, can I see the data that shows if our system is fastest?

Umm...



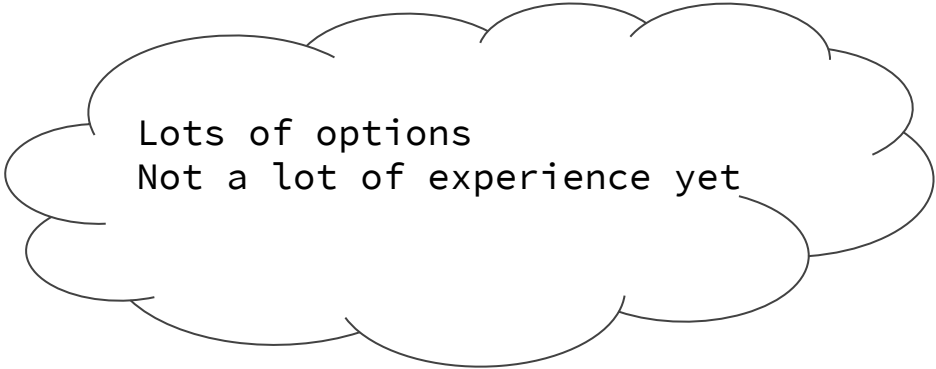
The (database) world at that time

NoSQL systems

- Google BigTable
- HBase
- Cassandra
- ...

Cloud systems

- PNUTS/Sherpa
- Amazon Dynamo
- ...



Lots of options
Not a lot of experience yet

The (Yahoo!) world at that time

Existing scalable but inconsistent storage systems

We were building PNUTS/Sherpa, but other parts of Yahoo were considering HBase and Cassandra

We were mere researchers, with no ability to force anybody to use our system

So we turned to science!

Funny story #2

How do you scale up a Yahoo user database?

How do we figure out if our system is faster?

Traditional answer: TPC-something

- But these were “NoSQL” systems!
- Also, the workloads were different

New answer: Write a blog post

- But hard to compare one blog post to another

What even is the question?

Fast at what?

- Reads? Writes?
- Large scans? Point operations?
- Throughput/latency? Scalability? Elasticity?

Our answer

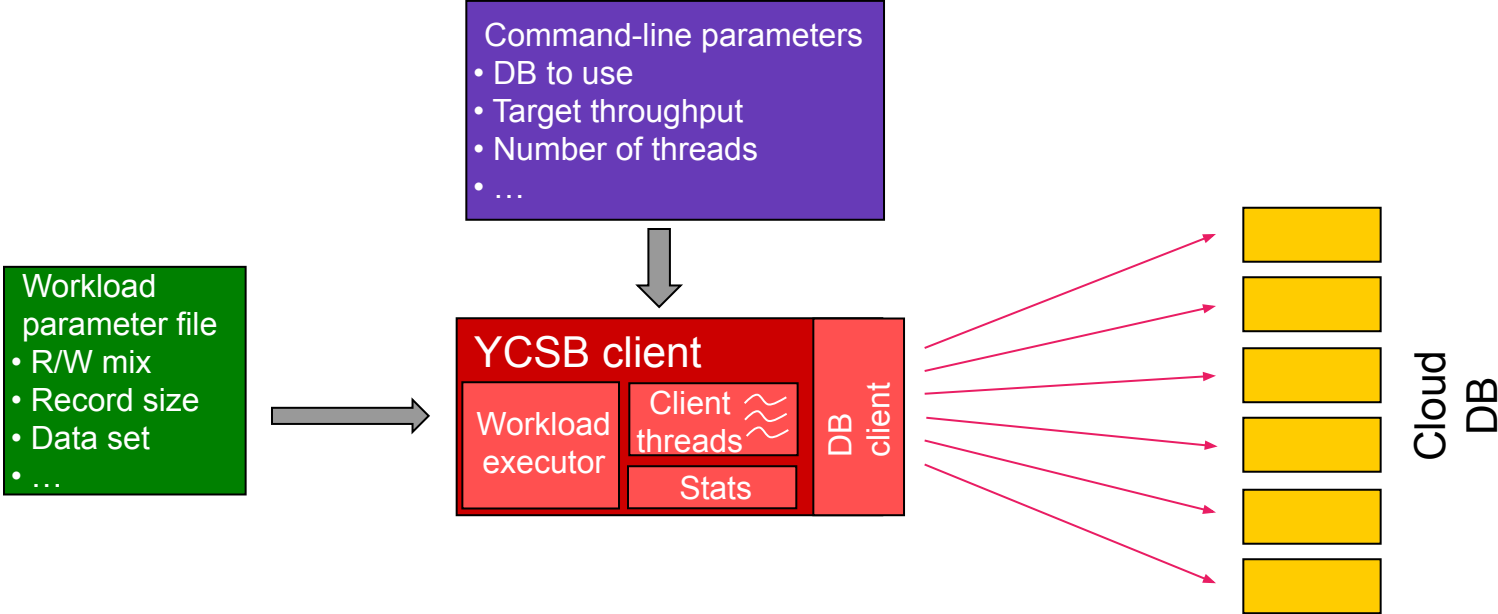
We wanted to:

- Define some workloads approximating what a web serving system would need
- Put the same workloads on multiple systems
- Draw some pretty graphs

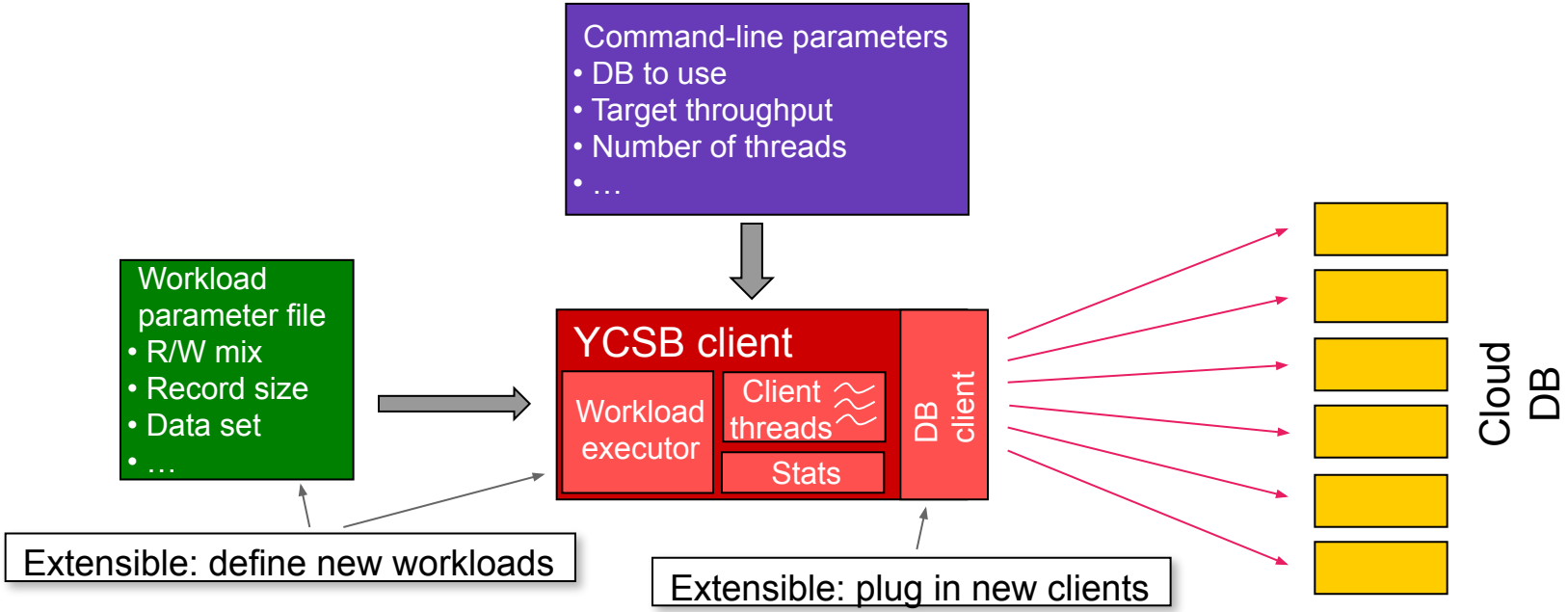
The result:

- Yahoo! Cloud Serving Benchmark (YCSB)

Benchmark tool



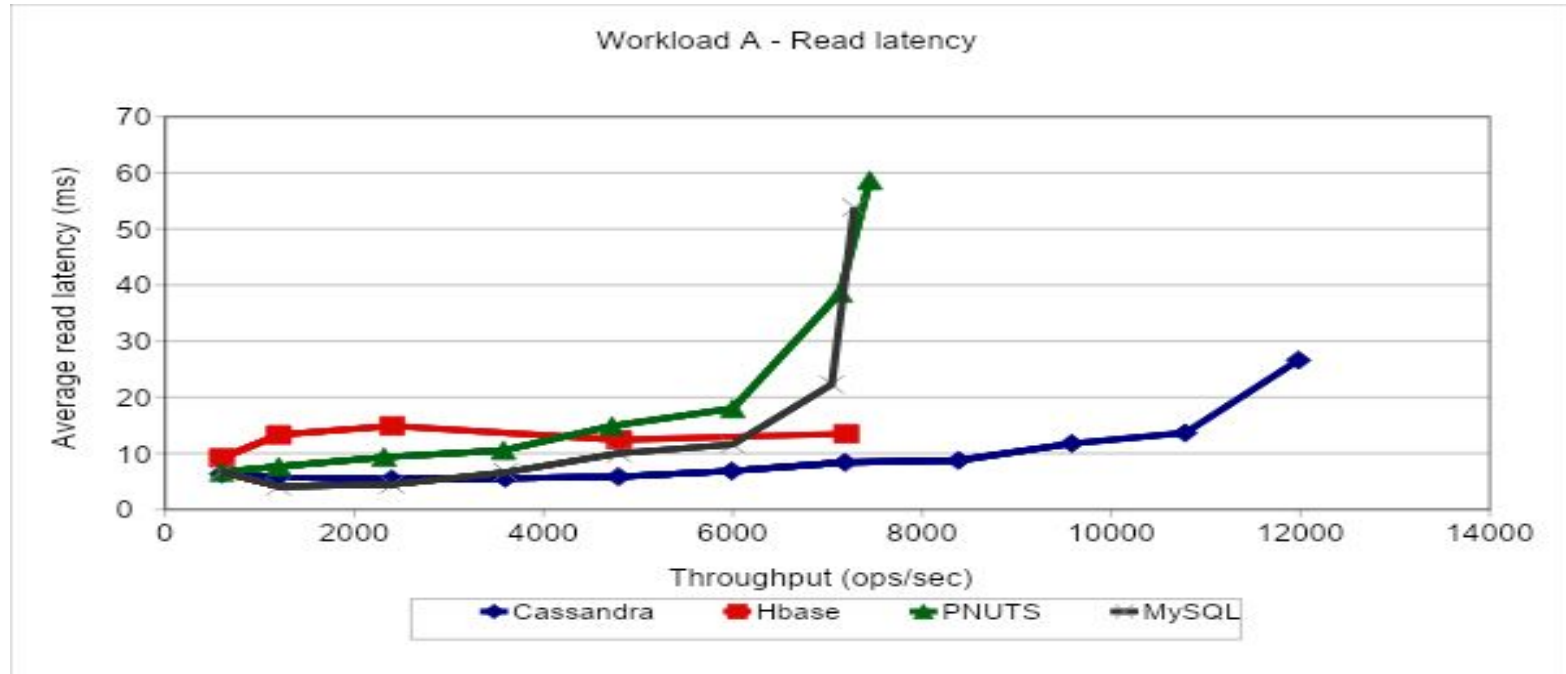
Benchmark tool



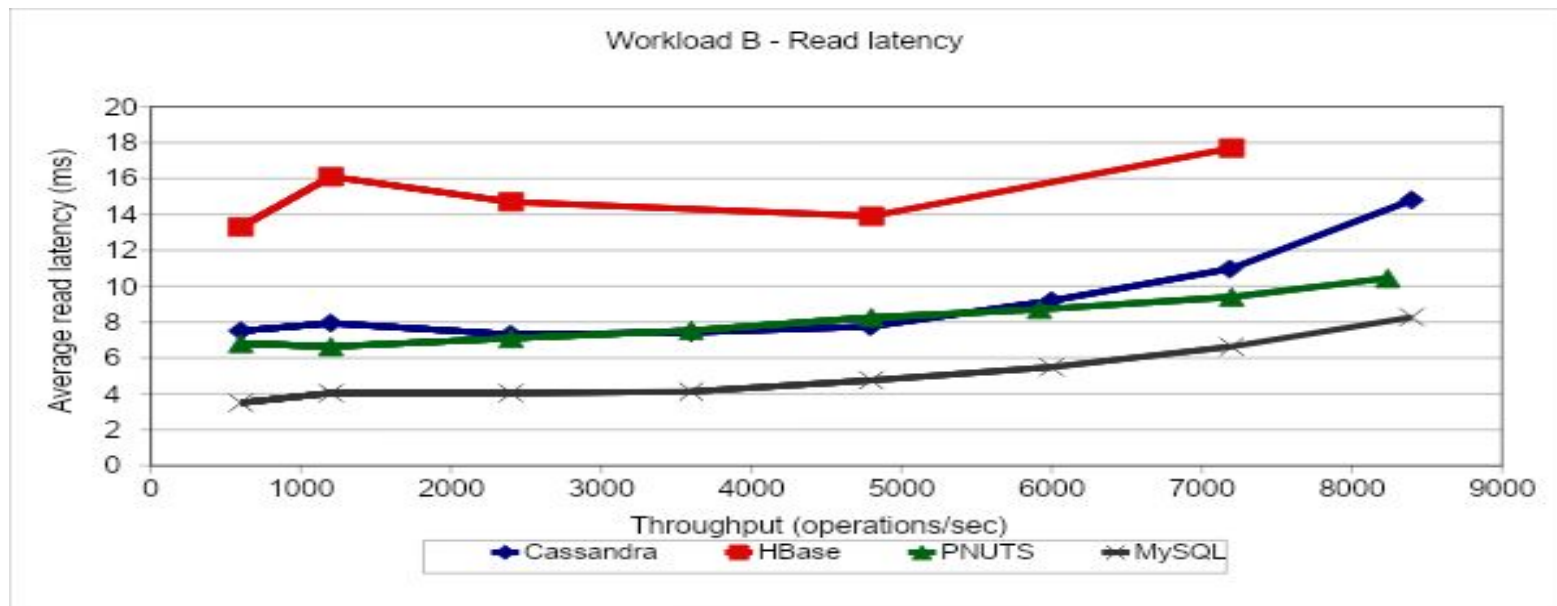
Workloads

- A – Update heavy
 - Session store
- B – Read heavy
 - Photo tagging
- C – Read only
 - Serving user profiles
- D – Read latest
 - User status updates
- E – Short ranges
 - Threaded conversations
- F – Read-modify-write
 - User metadata store

Sample results: Workload A (write heavy)



Sample results: Workload B (read heavy)



Lessons learned

Tools can be as valuable as new techniques or systems

Funny story #3

We wrote a paper, but it's the tool that had the impact

 [brianfrankcooper](#) / [YCSB](#)

 Watch 209

 Star 2.8k

 Fork 1.6k

 Code

 Issues 119

 Pull requests 44

 Projects 0

 Wiki

 Security

 Insights

<https://github.com/brianfrankcooper/YCSB>

The key to our success

Make it open source, easily extensible, and super easy to get results

Lessons learned

Not everybody is motivated by scientific inquiry

Lessons learned

Not everybody is motivated by scientific inquiry

- Or -

Researchers don't necessarily understand industry

Funny story #4

So I wrote this email to the HBase developer list...



Hi everybody! My name is Brian and I'm new here and I thought you'd like to know we benchmarked your system and it's pretty slow.

Funny story #4

The reaction was not positive

- They had their own measurements that showed HBase was very fast
- They thought we were a big corporation trying to ruin their open source project

For us, it was “just” a research project. For them, it was a fight for their project’s survival

Luckily...

We all got in a room and made nice and became friends

- They helped us tune their system to get better results
- They shipped some improvements to make their system faster
- We helped pick apart the distinction between scan and point workloads

Since that day...

Support for ~50 different backends

Widely used as an research experiment framework and as a commercial system benchmark

Managed by a great team of maintainers

- Sean Busbey, Andy Kruth, Eugene Blikh, Connor McCoy, Allan Bank, Chris Larsen, Chrisjan Matser, Govind Kamat, Kevin Risten, Jason Tedor, Stanley Feng

Funny story #5

All of the authors have worked at Google...

... except Raghu

(Someday we'll get him.)

Conclusion

A little science was needed at that time

We made it easy to measure Cloud (serving storage) systems

We are thankful for:

- Yahoo engineers who helped us run benchmarks
- Open source maintainers who have kept the tool going strong
- All the users!

Thanks!
