

All data stores mentioned in the book "Designing Data-Intensive Applications"



Irio Musskopf [Follow](#)

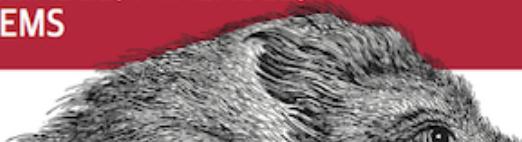
Sep 5, 2018 · 7 min read

When looking for good references for improving my software architecture skills, I came to the book "[Designing Data-Intensive Applications](#)," written by [Martin Kleppmann](#). As soon as I read the last page, I did a simple exercise: **tried to recall the databases mentioned throughout the previous 624 pages**. Checking personal notes or the book itself was strictly forbidden.

O'REILLY®

Designing Data-Intensive Applications

THE BIG IDEAS BEHIND RELIABLE, SCALABLE,
AND MAINTAINABLE SYSTEMS





Martin Kleppmann

<https://dataintensive.net/>

Since I could easily remember more than 20 products, my immediate conclusion was that I needed to **narrow down the studies**. Before trying to understand what could be useful in my future projects, I was forced to come up with methods for choosing a focus. Maybe the most cited technologies? That's when I remembered one of the most straightforward but useful applications of Apache Spark: **counting words!**

Methodology

I converted the Kindle book (purchased through Amazon.com) to a `.txt` file and loaded the contents into an Apache Spark server using Python. After experimenting with a couple of other strategies (most frequent capitalised words, TF-IDF), I selected the Index section and selected capitalised expressions starting new lines.

The outcome was a list of 342 words, which were verified manually for taking expressions such as “R-trees” and “ETL” out of the results. Since this job would be forcing me to recall the meaning of each name and search for official websites when still in question, I decided not to try to write an automation script.

Once the **list was narrowed to 72 items**, a straightforward word counter did the job. For every product with more than two words, I queried the book for the single most meaningful word. e.g., “Apache Kafka” refers to the number of times “Kafka” is mentioned. “(IBM) System R” had to be considered a single expression for not mixing with other kinds of system's. “(Google) Bigtable”, in the book, sometimes refer to the “Bigtable” data model, first proposed by Google's database and later implemented in other products. In the end, I decided to count both cases in favor of Google's product.

. . .

In a few cases, it's hard to draw a simple line of what is a data store and what is not. Apache Lucene, a dependence of both Elasticsearch and Apache Solr, was also added to the list.

. . .

(46) Apache ZooKeeper means that **ZooKeeper** is mentioned **46 times** in the book (without counting the Index section).

. . .

None of the logos are owned or were created by me, so I don't take responsibility over their eccentric design.

. . .

(46) Apache ZooKeeper



Apache ZooKeeper™



<https://zookeeper.apache.org/>

(44) PostgreSQL



<https://www.postgresql.org/>

(42) MySQL



<https://www.mysql.com/>

(41) Apache Kafka





<https://kafka.apache.org/>

(40) Apache Cassandra



<https://cassandra.apache.org/>

(37) Oracle Database





DATA BASE

<https://www.oracle.com/database/index.html>

(33) MongoDB



<https://www.mongodb.com/>

(31) Riak



<http://basho.com/products/>

(28) Apache HBase



<https://hbase.apache.org/>

(20) Microsoft SQL Server

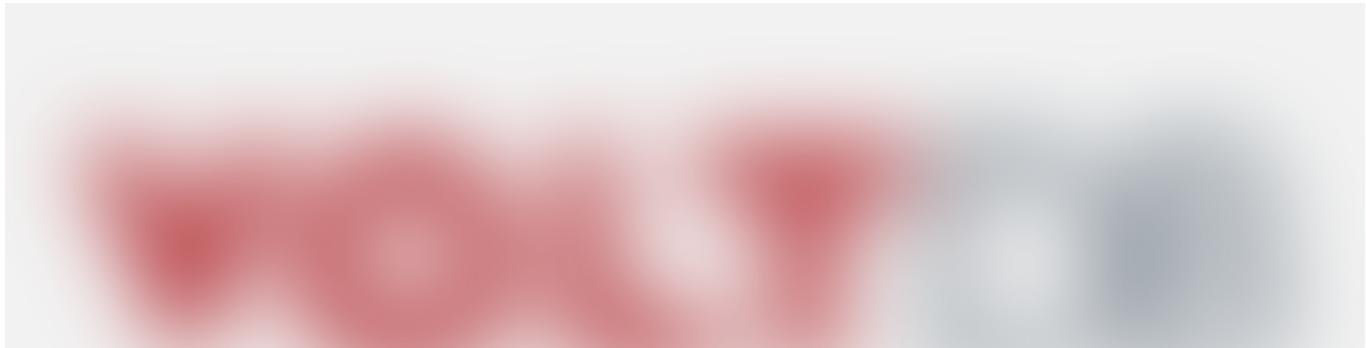




Microsoft® SQL Server®

<https://www.microsoft.com/en-us/sql-server>

(19) VoltDB



<https://www.voltdb.com/>

(17) Amazon DynamoDB



DynamoDB



<https://aws.amazon.com/dynamodb/>

(14) Apache Lucene



<https://lucene.apache.org/>

(14) Project Voldemort

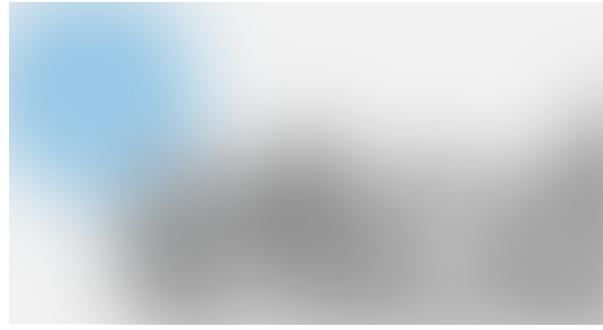


<http://www.project-voldemort.com/voldemort/>

(13) Apache CouchDB



(13) etcd



<https://coreos.com/etcd/>

(13) Datomic



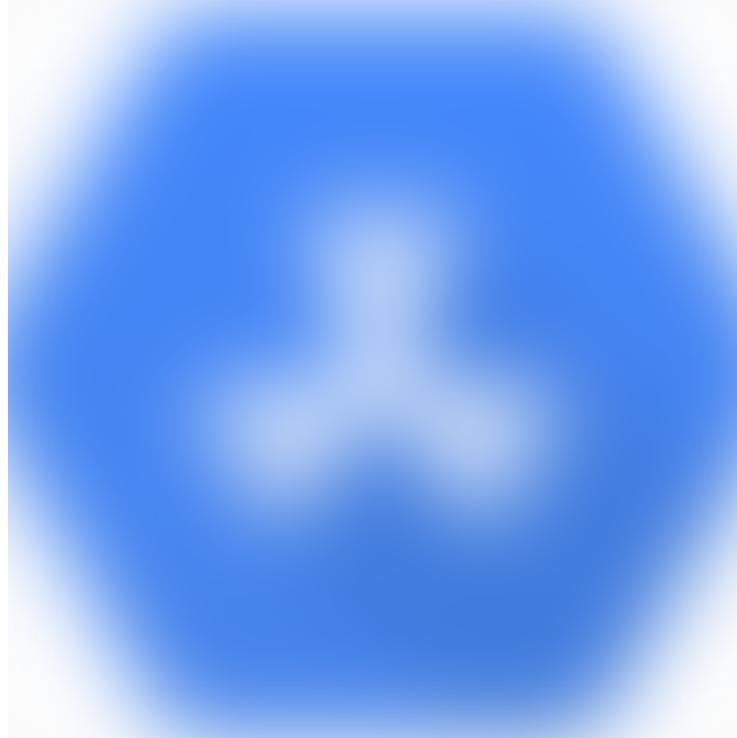
<https://www.datomic.com/>

(12) IBM Db2



<https://www.ibm.com/analytics/us/en/db2/>

(11) Google Spanner



<https://cloud.google.com/spanner/>

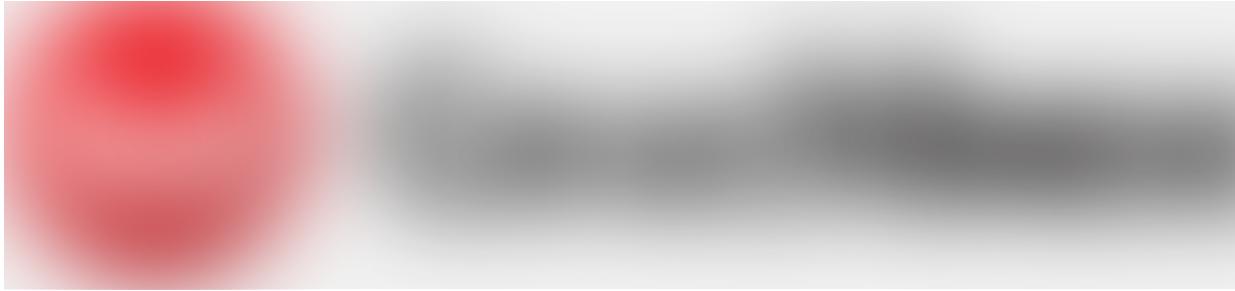
(10) Elasticsearch



elasticsearch

<https://www.elastic.co/products/elasticsearch>

(9) Couchbase Server



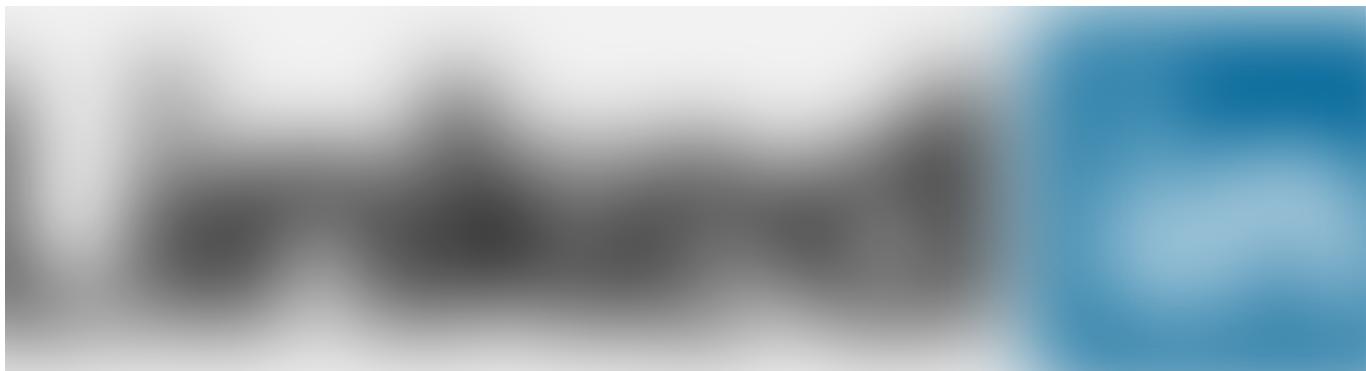
<https://www.couchbase.com/>

(9) Redis



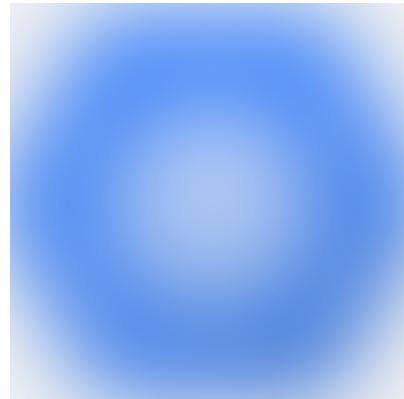
<https://redis.io/>

(8) LinkedIn Espresso



<https://engineering.linkedin.com/teams/data/projects/espresso>

(8) Google Bigtable



<https://cloud.google.com/bigtable/>

(8) RethinkDB



<https://www.rethinkdb.com/>

(8) LevelDB



<http://leveldb.org/>

(8) IBM IMS



<https://www.ibm.com/it-infrastructure/z/ims>

(7) IBM System R





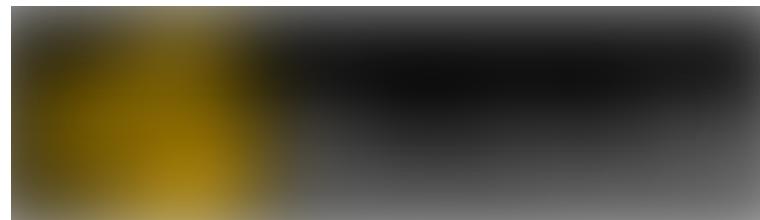
http://www.mcjones.org/System_R/

(6) Apache Solr



<https://lucene.apache.org/solr/>

(5) RocksDB



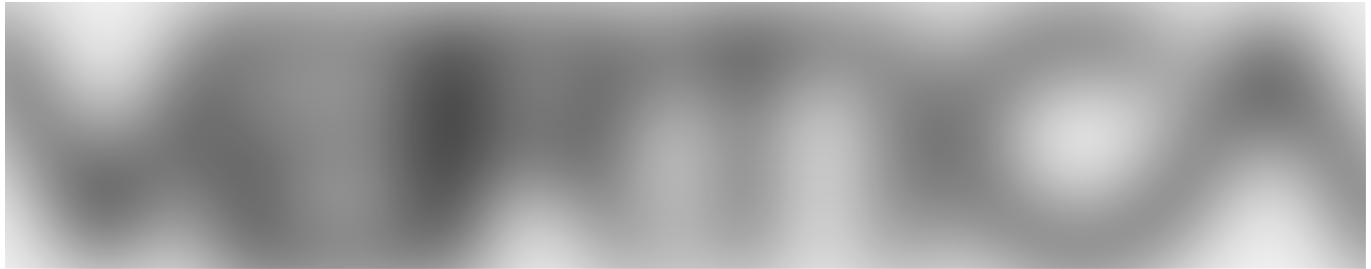
<https://rocksdb.org/>

(4) RabbitMQ



<https://www.rabbitmq.com/>

(4) Vertica



<https://www.vertica.com/>

(3) Microsoft Azure Storage



<https://azure.microsoft.com/en-us/services/storage/>

(3) Event Store



<https://eventstore.org/>

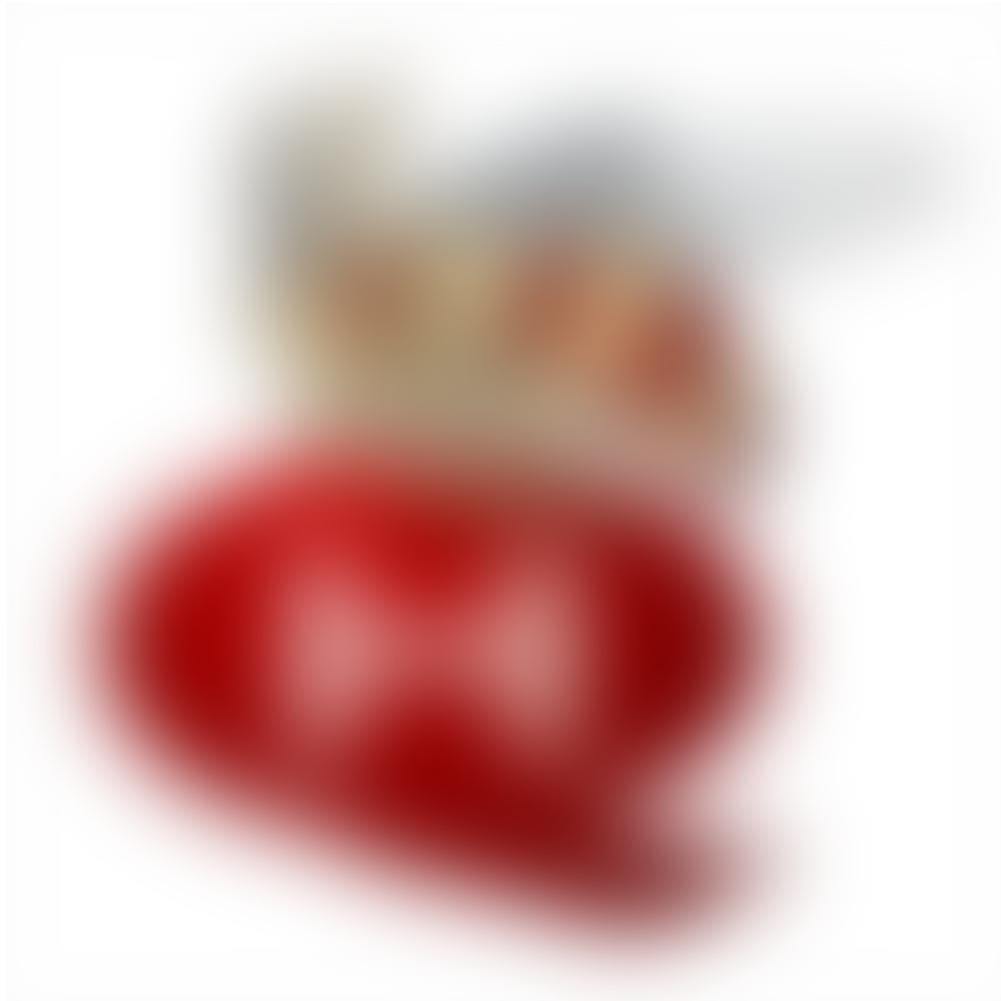
(3) SAP HANA





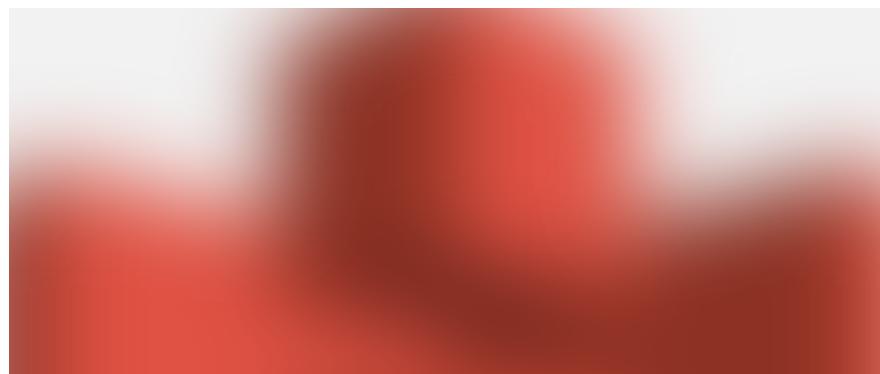
<https://www.sap.com/products/hana.html>

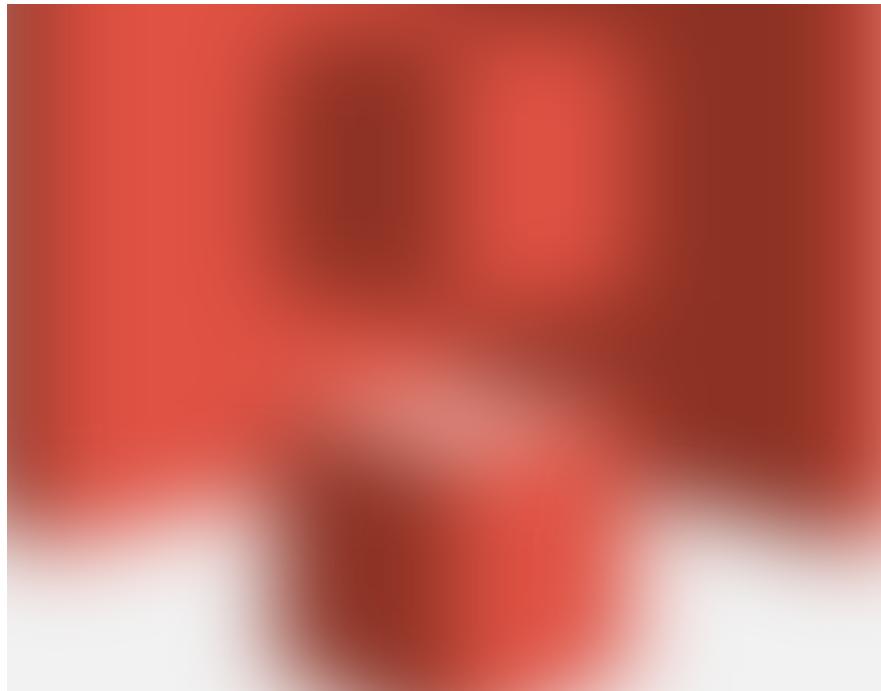
(3) HornetQ



<https://hornetq.jboss.org/>

(3) Amazon S3





<https://aws.amazon.com/s3/>

(3) Neo4j



<https://neo4j.com/>

(3) Apache DistributedLog





<https://bookkeeper.apache.org/distributedlog/>

(3) Apache ActiveMQ



<https://activemq.apache.org/>

(3) Memcached



<https://www.memcached.org/>

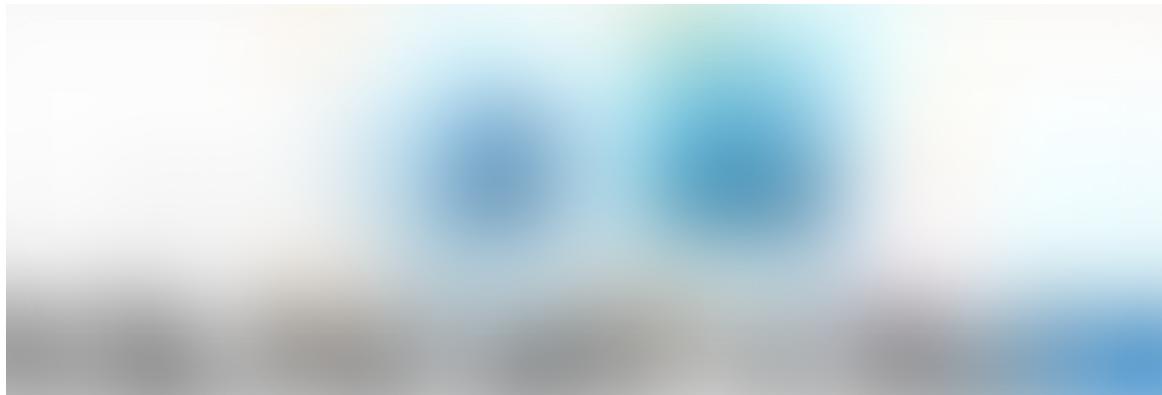
(3) Teradata





<https://www.teradata.co.uk/>

(3) FoundationDB



<https://www.foundationdb.org/>

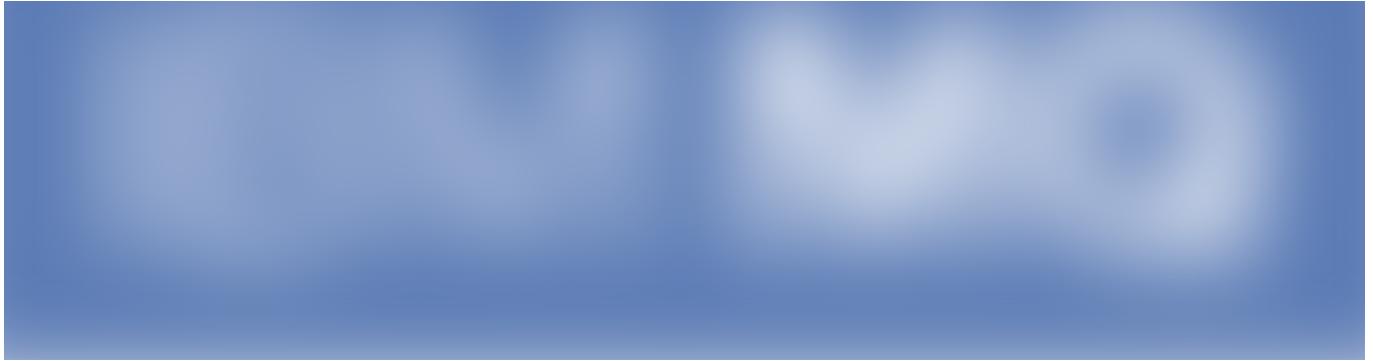
(3) Bayou



<https://github.com/HugoTian/Bayou>

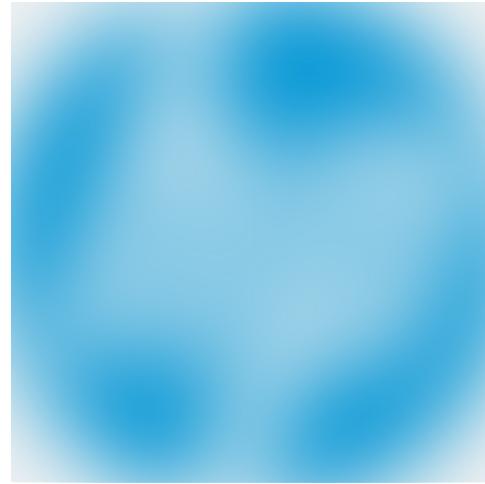
(2) IBM MQ





<https://developer.ibm.com/messaging/ibm-mq/>

(2) NonStop SQL



https://en.wikipedia.org/wiki/NonStop_SQL

(2) StatsD



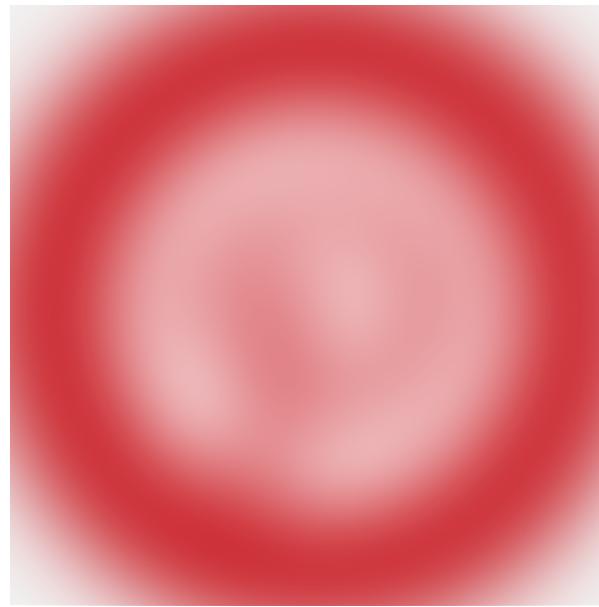
<https://github.com/etsy/statsd>

(2) HyperDex



<http://hyperdex.org/>

(2) Terrapin



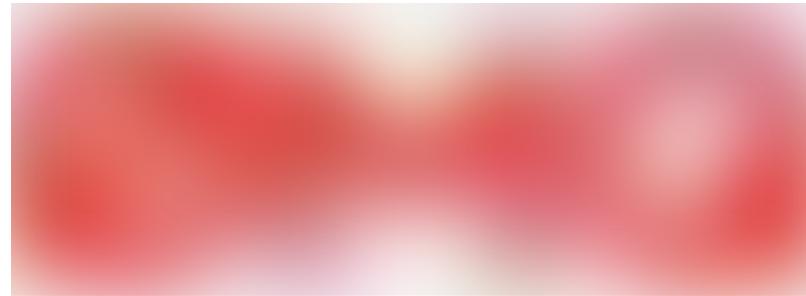
<https://github.com/pinterest/terrapin>

(2) Yahoo! Pistachio



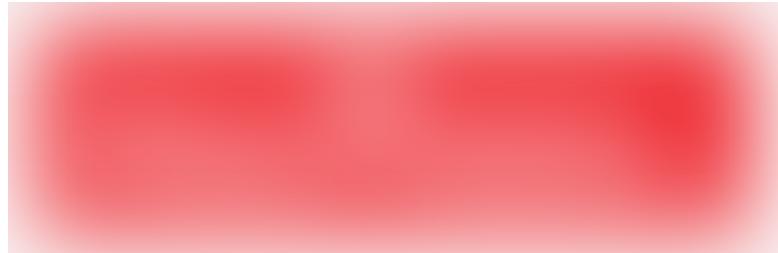
<https://github.com/lyogavin/Pistachio>

(2) ZeroMQ



<http://zeromq.org/>

(2) ParAccel



<http://www.paraccel.com/>

(2) Brubeck



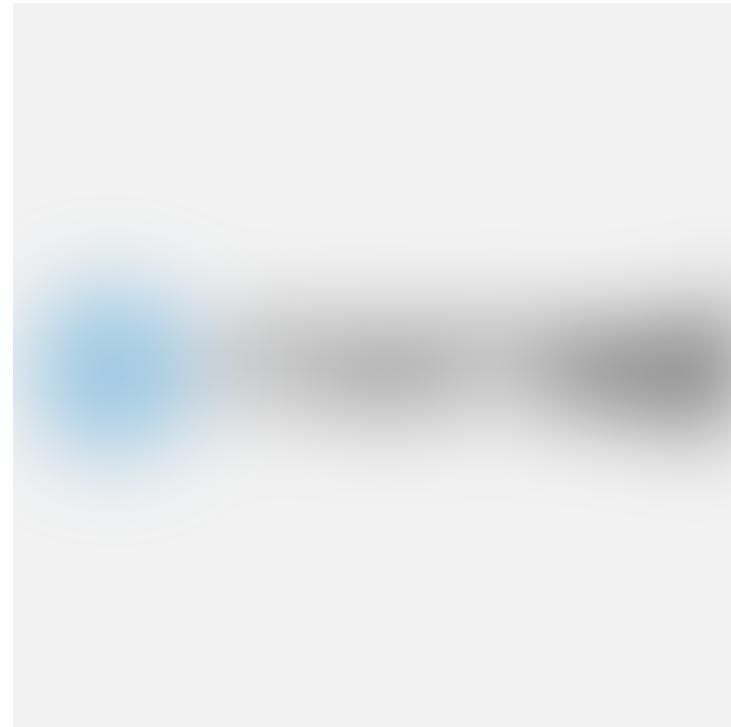
<https://github.com/jamesdabbs/brubeck.py>

(2) LMDB



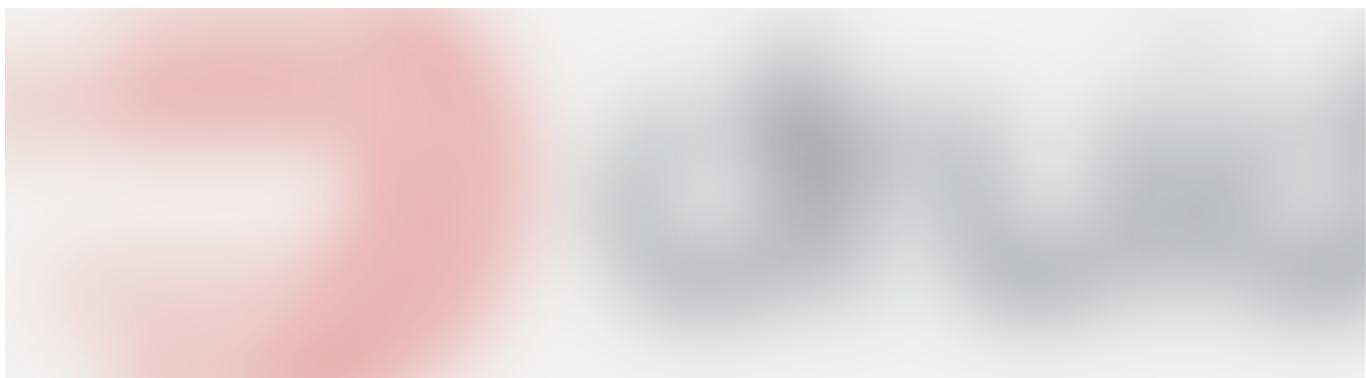
<http://www.lmdb.tech/doc/>

(2) MemSQL



<https://www.memsql.com/>

(2) Druid



<http://druid.io/>

(2) Consul



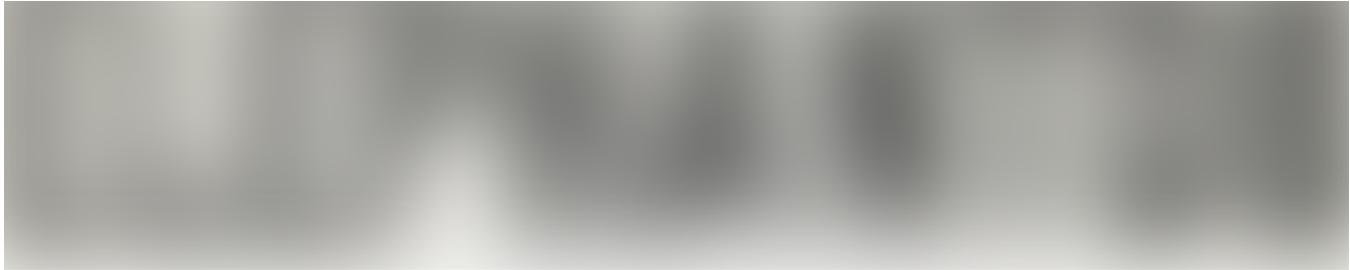
<https://www.consul.io>

(2) Firebase



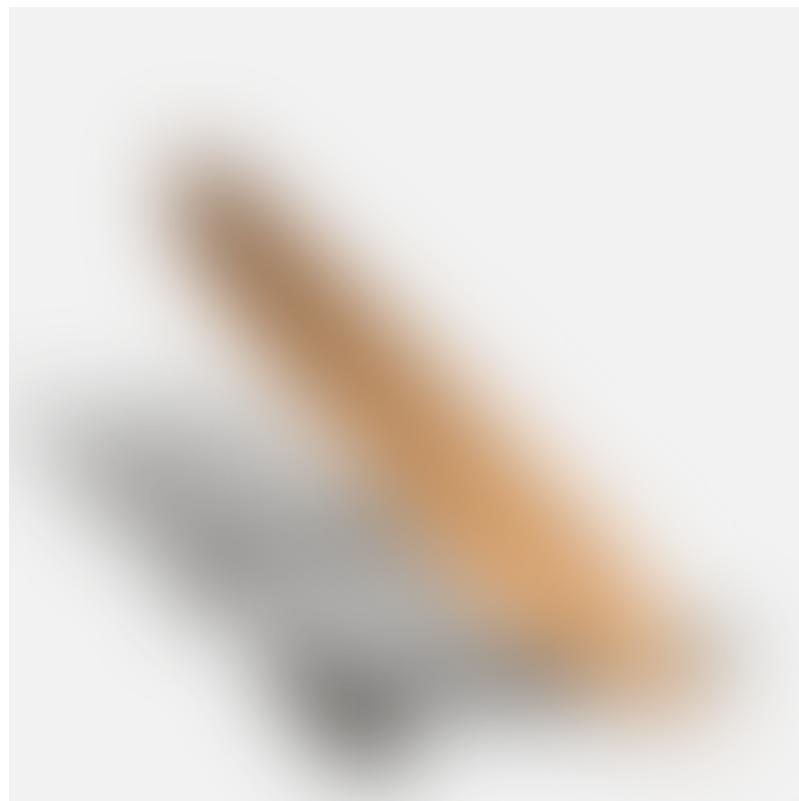
<https://firebase.google.com/docs/database/>

(2) ElephantDB



<https://github.com/nathanmarz/elephantdb>

(2) Apache BookKeeper



<https://bookkeeper.apache.org/>

(2) IBM WebSphere



<https://www.ibm.com/cloud/websphere-application-platform>

(1) Microsoft Azure Service Bus



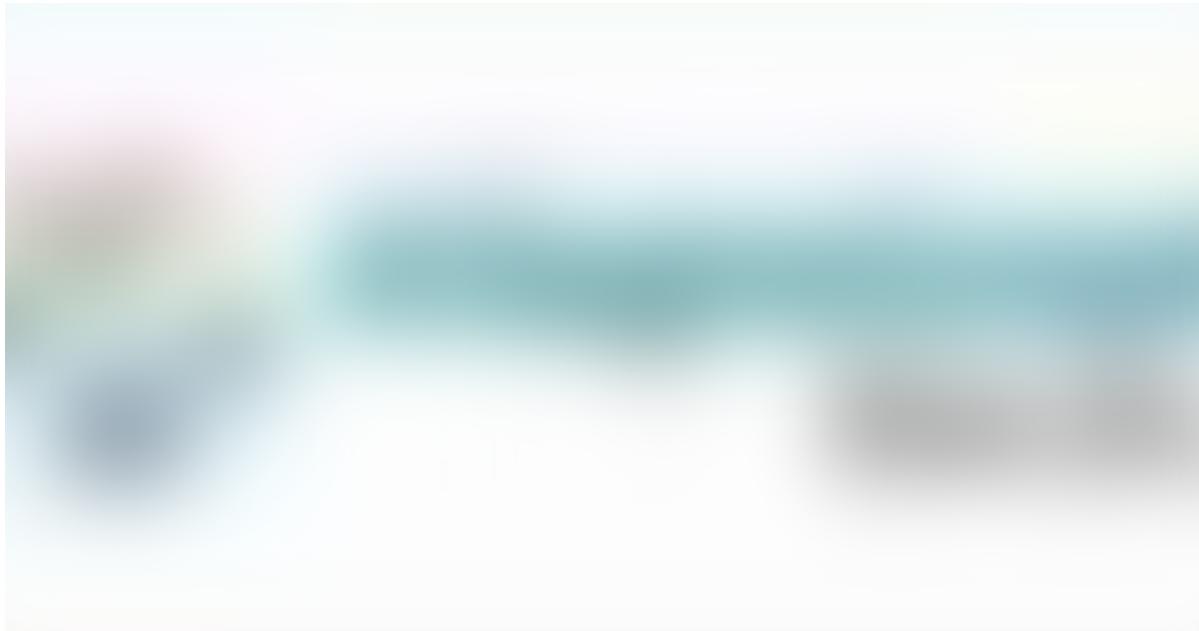
<https://azure.microsoft.com/en-us/services/service-bus/>

(1) Apache Qpid



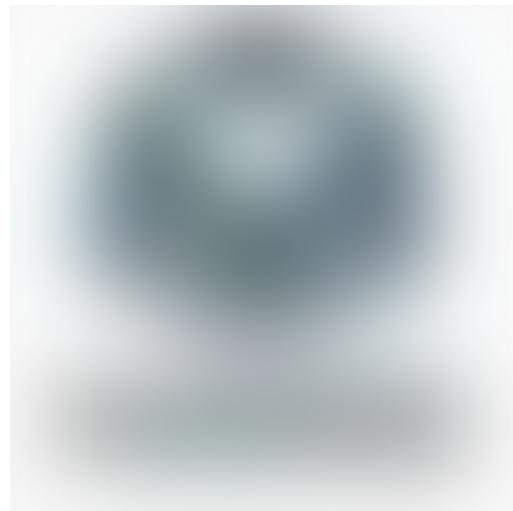
<https://qpid.apache.org/>

(1) AllegroGraph



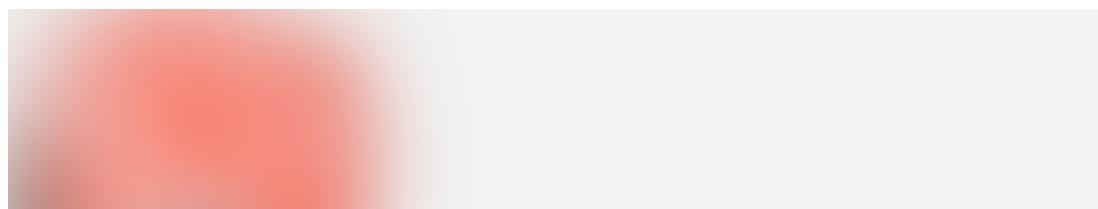
<https://franz.com/agraph/allegrograph/>

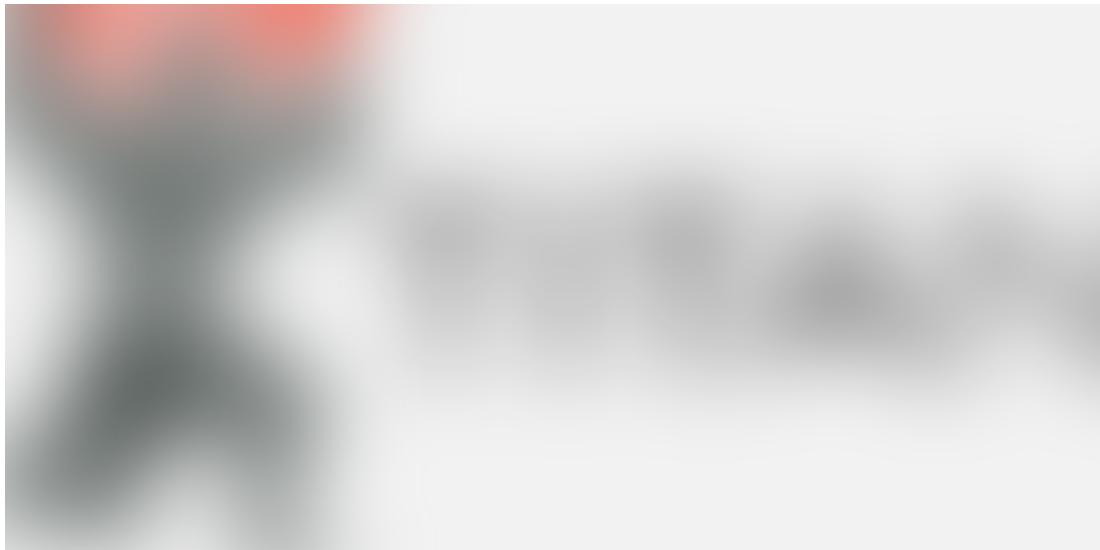
(1) Apache HAWQ



<http://hawq.apache.org/>

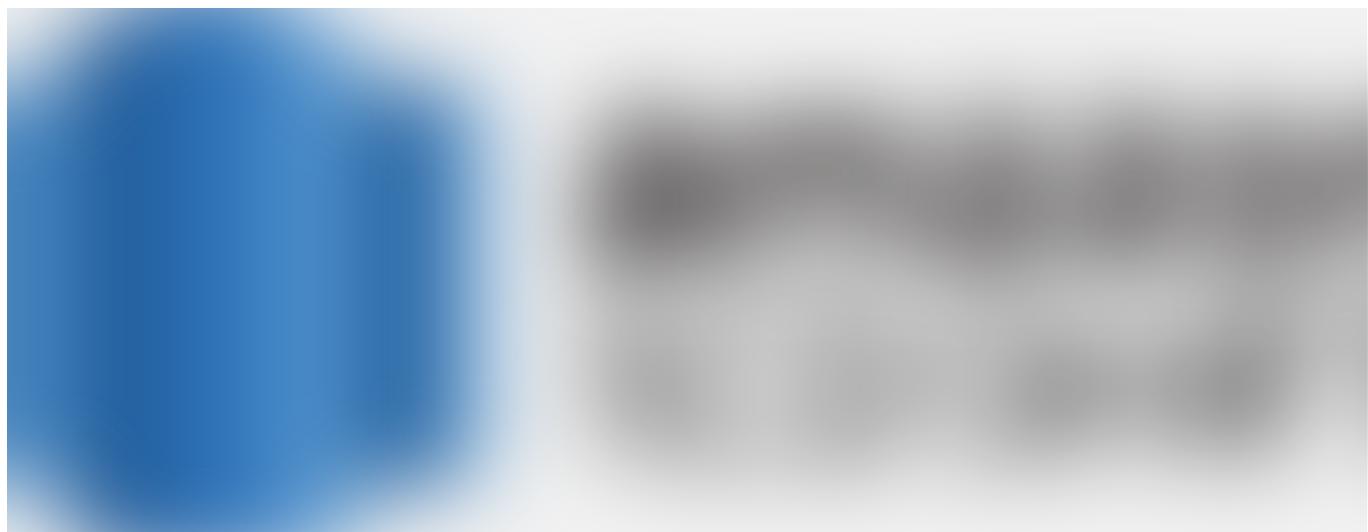
(1) Titan





<https://titan.thinkaurelius.com/>

(1) Amazon RedShift



<https://aws.amazon.com/redshift/>

(1) InfiniteGraph



<https://www.objectivity.com/products/infinitegraph/>

(1) NATS



<https://nats.io/>

(1) RAMCloud



<https://ramcloud.atlassian.net/wiki/spaces/RAM/overview?mode=global>

(1) MSMQ



<https://msdn.microsoft.com/en-us/library/ms711472%28v=vs.85%29.aspx?f=255&MSPPError=-2147217396>

(1) webMethods





https://www.softwareag.com/corporate/products/webmethods_integration/application_integration/webmethods_adapters.html

• • •

If it isn't obvious yet, yes, you should get yourself a copy of Martin's book. **Well worth the investment of time and money.**

Some rights reserved 

Big Data

Software Engineering

Apache Spark

Database

Data Analysis

Medium

About Help Legal