

Difference between scaling horizontally and vertically for databases [closed]

Asked 12 years, 4 months ago Modified 2 years, 10 months ago

Viewed 473k times



815



Closed. This question needs to be more [focused](#). It is not currently accepting answers.



Want to improve this question? Update the question so it focuses on one problem only by [editing this post](#).

Closed 4 years ago.

[Improve this question](#)

I have come across many NoSQL databases and SQL databases. There are varying parameters to measure the strength and weaknesses of these databases and scalability is one of them. What is the difference between horizontally and vertically scaling these databases?

database

database-design

nosql

scalability

Share Follow

asked Jul 29, 2012 at 8:40



London guy

28k ● 46 ● 126 ● 182

2 en.wikipedia.org/wiki/Scalability - the term applies to all software/systems – Tomasz Nurkiewicz Jul 29, 2012 at 8:41



5 Pay special attention to the *Database* section
en.wikipedia.org/wiki/Scalability#Database_scalability
– user454322 Aug 15, 2014 at 3:41

2 [stackoverflow.com/questions/5401992/...](https://stackoverflow.com/questions/5401992/)
– Ciro Santilli OurBigBook.com Sep 11, 2015 at 15:02

10 Answers

Sorted by:

Highest score (default)



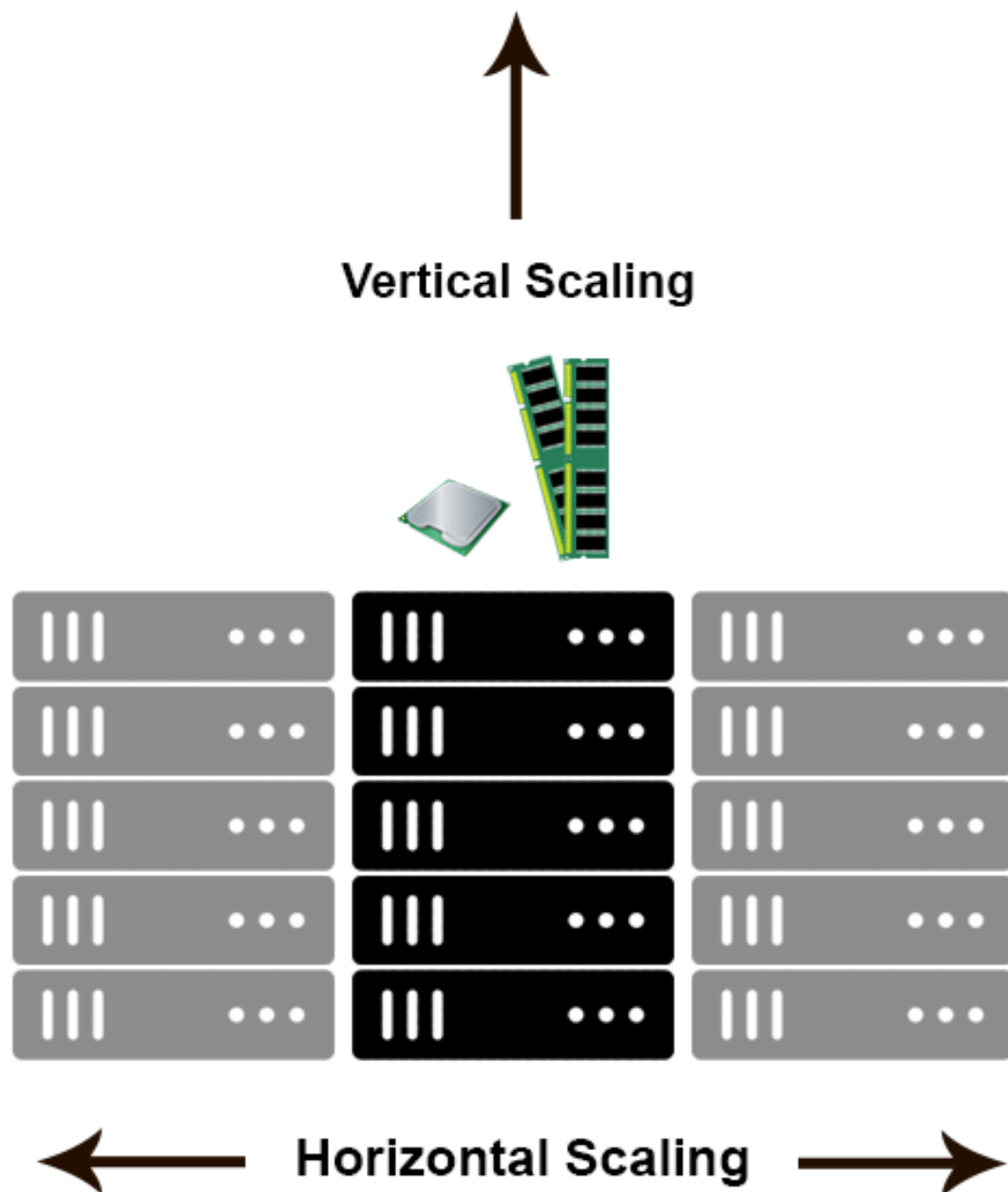
1434

Horizontal scaling means that you scale by adding more machines into your pool of resources whereas **Vertical scaling means that you scale by adding more power (CPU, RAM) to an existing machine.**



An easy way to remember this is to think of a machine on a server rack, we add more machines across the **horizontal** direction and add more resources to a machine in the **vertical** direction.





In the database world, horizontal-scaling is often based on the partitioning of the data i.e. each node contains only part of the data, in vertical-scaling the data resides on a single node and scaling is done through multi-core i.e. spreading the load between the CPU and RAM resources of that machine.

With horizontal-scaling it is often easier to scale dynamically by adding more machines into the existing pool - Vertical-scaling is often limited to the capacity of a

single machine, scaling beyond that capacity often involves downtime and comes with an upper limit.

Good examples of horizontal scaling are Cassandra, MongoDB, [Google Cloud Spanner](#) .. and a good example of vertical scaling is MySQL - Amazon RDS (The cloud version of MySQL). It provides an easy way to scale vertically by switching from small to bigger machines. This process often involves downtime.

In-Memory Data Grids such as [GigaSpaces XAP](#), [Coherence](#) etc.. are often optimized for both horizontal and vertical scaling simply because they're not bound to disk. Horizontal-scaling through partitioning and vertical-scaling through multi-core support.

You can read more on this subject in my earlier posts: [Scale-out vs Scale-up](#) and [The Common Principles Behind the NOSQL Alternatives](#)

Share Follow

edited Aug 29, 2020 at 21:46

community wiki
9 revs, 9 users 59%
Nati Shalom

1 There is also Couchbase, Riak, HBase, CitrusLeaf and Infinispan to complete the list a bit further (there are more).
– [scalabl3](#) Aug 1, 2012 at 6:39

12 i am kind of confused here ... adding more machines is effectively same to adding more cpu/ram .. then how the two

of them are different because when we add a new machine it comes with cpu and ram , please correct me if i am wrong.

– [Subham Tripathi](#) May 16, 2015 at 5:01

- 14 @SubhamTripathi As explained here, vertical scaling is limited to one server (or a small group of servers) and it has a practical upper limit (meaning you can't go beyond say 512 GB of RAM). Horizontal scaling on the other hand can practically happen indefinitely. – [asgs](#) Jun 9, 2015 at 6:56
-

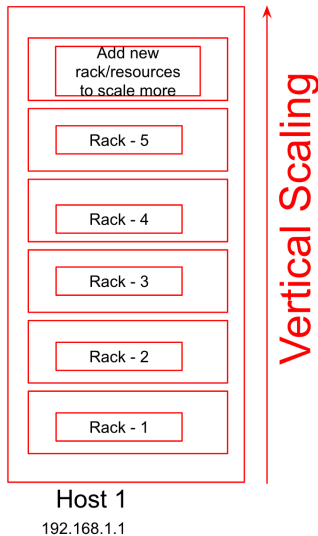


Scaling horizontally ==> Thousands of minions will do the work together for you.

274



Scaling vertically ==> One big hulk will do all the work for you.



To scale more, Add more RAM, CPU, Memory to the **one existing machine**

Horizontal Scaling

To scale more: Add more machines to existing **group of distributed system**



Share Follow

edited Oct 26, 2020 at 18:17

answered Oct 25, 2018 at 9:43



Sankarganesh Eswaran

10.3k ● 3 ● 25 ● 24



29

Let's start with the need for scaling that is increasing resources so that your system can now handle more requests than it earlier could.



When you realise your system is getting slow and is unable to handle the current number of requests, you need to scale the system.





This provides you with two options. Either you increase the resources in the server which you are using currently, i.e, increase the amount of RAM, CPU, GPU and other resources. This is known as vertical scaling.

Vertical scaling is typically costly. It does not make the system fault tolerant, i.e if you are scaling application running with single server, if that server goes down, your system will go down. Also the amount of threads remains the same in vertical scaling. Vertical scaling may require your system to go down for a moment when process takes place. Increasing resources on a server requires a restart and put your system down.

Another solution to this problem is increasing the amount of servers present in the system. This solution is highly used in the tech industry. This will eventually decrease the request per second rate in each server. If you need to scale the system, just add another server, and you are done. You would not be required to restart the system. Number of threads in each system decreases leading to high throughput. To segregate the requests, equally to each of the application server, you need to add load balancer which would act as reverse proxy to the web servers. This whole system can be called as a single cluster. Your system may contain a large number of requests which would require more amount of clusters like this.

Hope you get the whole concept of introducing scaling to the system.

Share Follow

edited Nov 28, 2018 at 2:27



Pang

10.1k ● 146 ● 85 ● 124

answered Feb 13, 2018 at 7:41



yathartha

492 ● 5 ● 8



11



There is an additional architecture that wasn't mentioned - SQL-based database services that enable horizontal scaling without the complexity of manual sharding. These services do the sharding in the background, so they enable you to run a traditional SQL database and scale out like you would with NoSQL engines like MongoDB or CouchDB. Two services I am familiar with are [EnterpriseDB](#) for PostgreSQL and [Xeround](#) for MySQL. I saw an in-depth [post](#) by Xeround which explains why scale-out on SQL databases is difficult and how they do it differently - treat this with a grain of salt as it is a vendor post. Also check out Wikipedia's [Cloud Database entry](#), there is a nice explanation of SQL vs. NoSQL and service vs. self-hosted, a list of vendors and scaling options for each combination. ;)

Share Follow

answered Sep 10, 2012 at 9:52



Dina Kaiser

476 ● 3 ● 6

As another data point, I submit another vendor post from Clustrix: clustrix.com/blog/bid/259950/scale-up-vs-scale-out
– [clieu](#) Jan 26, 2013 at 2:12 ✎

- 1 I know this is an old post... just some updates .. Xeround has closed shop. PostgreSQL's horizontal scaling options are not really horizontal scaling options - they are just DB replication options where you can spawn out some operations to the replicated DB. – [Dharmendar Kumar 'DK'](#) Jan 11, 2018 at 20:04
-



9



Yes scaling horizontally means adding more machines, but it also implies that the machines are equal in the cluster. MySQL can scale horizontally in terms of Reading data, through the use of replicas, but once it reaches capacity of the server mem/disk, you have to begin sharding data across servers. This becomes increasingly more complex. Often keeping data consistent across replicas is a problem as replication rates are often too slow to keep up with data change rates.

Couchbase is also a fantastic NoSQL Horizontal Scaling database, used in many commercial high availability applications and games and arguably the highest performer in the category. It partitions data automatically across cluster, adding nodes is simple, and you can use commodity hardware, cheaper vm instances (using Large instead of High Mem, High Disk machines at AWS for instance). It is built off the Membase (Memcached) but adds persistence. Also, in the case of Couchbase, every node can do reads and writes, and are equals in the cluster, with only failover replication (not full dataset replication across all servers like in mySQL).

Performance-wise, you can see an excellent Cisco benchmark: <http://blog.couchbase.com/understanding-performance-benchmark-published-cisco-and-solarflare-using-couchbase-server>

Here is a great blog post about Couchbase Architecture: <http://horicky.blogspot.com/2012/07/couchbase-architecture.html>

Share Follow

edited Aug 1, 2012 at 6:39

answered Aug 1, 2012 at 6:34



scalabl3

1,273 ● 6 ● 7

What do you mean by "using Large instead of High Mem" ?
large vs high memory? was that a typo or is there a
difference – [Alexandros Kourtis](#) Aug 29, 2020 at 21:46



8



Traditional relational databases were designed as client/server database systems. They can be scaled horizontally but the process to do so tends to be complex and error prone. NewSQL databases like NuoDB are memory-centric distributed database systems designed to scale out horizontally while maintaining the SQL/ACID properties of traditional RDBMS.

For more information on NuoDB, read their [technical white paper](#).

Share Follow

edited Jan 26, 2019 at 1:43



[Timo Tijhof](#)

10.3k ● 6 ● 37 ● 53

answered Jul 14, 2014 at 12:03



[Michael Waclawiczek](#)

81 ● 1 ● 1



5



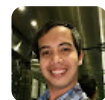
SQL databases like Oracle, db2 also support Horizontal scaling through Shared disk cluster. For example Oracle RAC, IBM DB2 purescale or Sybase ASE Cluster edition. New node can be added to Oracle RAC system or DB2 purescale system to achieve horizontal scaling.



But the approach is different from noSQL databases (like mongodb, CouchDB or IBM Cloudant) is that the data sharding is not part of Horizontal scaling. In noSQL databases data is sharded during horizontal scaling.

Share Follow

edited Sep 9, 2017 at 10:09



[Khoa TruongDinh](#)

933 ● 1 ● 13 ● 27

answered Jan 4, 2017 at 8:14



[Debasish](#)

51 ● 1 ● 1



5

The accepted answer is spot on the basic definition of horizontal vs vertical scaling. But unlike the common belief that horizontal scaling of databases is only possible with Cassandra, MongoDB, etc I would like to add that



horizontal scaling is also very much possible with any traditional RDMS; that too without using any third party solutions.



I know of many companies, specially SaaS based companies that do this. This is done using simple application logic. You basically take a set of users and divide them over multiple DB servers. So for example, you would typically have a "meta" database/table that would store clients, DB server/connection strings, etc and a table that stores client/server mapping.

Then simply direct requests from each client to the DB server they are mapped to.

Now some may say this is akin to horizontal partitioning and not "true" horizontal scaling and they will be right in some ways. But the end result is that you have scaled your DB over multiple Db servers.

The only difference between the two approaches to horizontal scaling is that one approach (MongoDB, etc) the scaling is done by the DB software itself. In that sense you are "buying" the scaling. In the other approach (for RDBMS horizontal scaling), the scaling is built by application code/logic.

Share Follow

edited Feb 24, 2022 at 21:05

answered Sep 15, 2019 at 20:06



Dharmendar Kumar 'DK'

2,102 ● 17 ● 18



0



Adding lots of load balancers creates extra overhead and latency and that is the drawback for scaling out horizontally in nosql databases. It is like the question why people say RPC is not recommended since it is not robust.

I think in a real system we should use both sql and nosql databases to utilize both multicore and cloud computing capabilities of today's systems.

On the other hand, complex transactional queries has high performance if sql databases such as oracle being used. NoSql could be used for bigdata and horizontal scalability by sharding.

Share Follow

answered Apr 20, 2018 at 11:51



farshad-nsh

91 ● 1 ● 5



0



You have a company and there is only 1 worker but you got 1 new project at that time you hire new candidate -- this is horizontal scaling. where new candidate is new machines and project is new traffic/calls to your api's.

Where as 1 project with an IIT/NIT guy handling all request to your api/traffic. If any time more request to

your api's then fire him and replacing him with a high IQ NIT/IIT guy -- this is vertical scaling.

Share Follow

answered Nov 6, 2019 at 13:26



Jithin Kumar S

873 ● 2 ● 11 ● 23
