Why is it not advisable to have the database and web server on the same machine?

Listening to Scott Hanselman's interview with the Stack Overflow team (part 1 and 2), he was adamant that the SQL server and application server should be on separate machines. Is this just to make sure that if one server is compromised, both systems aren't accessible? Do the security concerns outweigh the complexity of two servers (extra cost, dedicated network connection between the two, more maintenance, etc.), especially for a small application, where neither piece is using too much CPU or memory? Even with two servers, with one server compromised, an attacker could still do serious damage, either by deleting the database, or messing with the application code.

Why would this be such a big deal if performance isn't an issue?

database security networking infrastructure hardware-infrastructure

asked Mar 18 '09 at 20:29
Tai Squared
6,833 ● 20 ● 64 ● 79

18 Answers

1. Security. Your web server lives in a DMZ, accessible to the public internet and taking untrusted input from anonymous users. If your web server gets compromised, and you've followed least privilege rules in connecting to your DB, the maximum exposure is what your app can do through the database API. If you have a business tier in between, you have one more step between your attacker and your data. If, on the other hand, your database is on the same server, the attacker now has root access to your data and server.
2. Scalability. Keeping your web server stateless allows you to scale your web servers horizontally pretty much effortlessly. It is *very* difficult to horizontally scale a database server.
3. Performance. 2 boxes = 2 times the CPU, 2 times the RAM, and 2 times the spindles for disk access.

All that being said, I can certainly see reasonable cases that none of those points really matter.

answered Mar 18 '09 at 20:48
Mark Brackett
71.7k ● 15 ● 90 ● 141

12 But with 2 machines you have double the probability of hardware failure ;) – TWith2Sugars Mar 19 '09 at 12:21

3 @TWith2Sugars - as opposed to what? – Kev Mar 19 '09 at 13:51

3 Ref. point 1. If the Web Tier is owned, then what more do you want or need than the App /database API interface? Surely it's game over at that point anyway? Things then get very interesting in terms of services required to support DMZ infrastructure e.g any AD or Microsoft services in play? – Noelie Dunne Mar 24 '09 at 14:37

4 @Noelie - Your web tier would not have access to say, backup the database to a file and ftp that file to ftp.hackers.com using xp_cmdshell. Or drop the database. Or modify config values. etc. – Mark Brackett Mar 24 '09 at 14:47

5 @kirgy - since the two machines are in parallel and are each a single point of failure, the overall reliability is less; it's not technically double (it's actually 1 - (r1 * r2)) but close enough for large reliability and small nodes....In the 0.01% case, it'd be 0.0199%. But for 100 nodes, it'd be 36.6% instead of the 100% implied by the doubling statement. – Mark Brackett Feb 24 '16 at 16:28

It doesn't *really* matter (you can quite happily run your site with web/database on the same machine), it's just the easiest step in scaling..

It's exactly what StackOverflow did - starting with single machine running IIS/SQL Server, then when it started getting heavily loaded, a second server was bought and the SQL server was moved onto that.

If performance is not an issue, do not waste money buying/maintaining two servers.

answered Mar 18 '09 at 21:44
dbr
109k ● 50 ● 244 ● 308

3 I agree, it can be done while the load is low...as the load increases, its easy to seperate them into 2 or more machines. – E.J. Brennan Mar 18 '09 at 21:56

4 I came in and was going to comment on the same thing. Switching you DB server is as easy as changing your connection string (in most cases). – CitizenBane Oct 5 '09 at 14:24

1 Totally true words – elkebirmed Mar 14 '14 at 17:02

on the same machine as the web site. However, in their customer's case, usually the db & web server are the only applications on that machine, and the website isn't straining the machine that much. Then, the efficiency of not having to send data across the network more that made up for the increased strain.

answered Mar 18 '09 at 20:39

James Curran
80.7k ● 28 ● 152 ● 232

2 ...until concurrent usage increases, and the db server needs more memory to make effective use of buffers and caches. Once the web/app and db servers need more memory than they can share on a single box, paging and disk I/O increases, and performance tanks. – MattK Mar 18 '09 at 21:35

@MattK - but what if you have massive amounts of memory? We have an app where each client has their own database, so both the database and web servers can scale horizontally very easily. Given that we have more memory than disk in use (64GB vs. ~40GB), wouldn't it be better for performance to keep it all on the same machine? – Beep beep May 8 '11 at 2:56

If your working set fits in memory, then you may not see any of the performance issues I mention. More often servers have more disk than RAM, but it sounds like in your case you have databases that will fit entirely in RAM - as long as the applications on the shared server do not consume too much of it. – MattK May 11 '11 at 21:46

I would think the big factor would be performance. Both the web server/app code and SQL Server would cache commonly requested data in memory and you're killing your cache performance by running them in the same memory space.

answered Mar 18 '09 at 20:33

Tom Ritter
77.2k ● 27 ● 122 ● 160

8 What if you have a small(ish) database, but with tons of memory? Wouldn't the cost of going over the network for each database call, particularly if there are many, outweigh the benefits? – Beep beep May 8 '11 at 2:53

1 @Tom Ritter Although performance is a concern (Per this answer, and point #3 in Mark Brackett's answer) I know that some people (me included) would likely put the money saved by NOT getting a separate DB server into MORE CPU/RAM/etc. on the one-and-only server that replaced it. So this should be taken into account. As for separated memory, IIS and SQL could be configured to account for their competition over resources. Personally, the "kicker" for me was Mark's #1 point... security. I like the thought of the limited access a compromised webserver would have to a *separate* DB server. – Dylan - INNO Software Oct 9 '12 at 6:02

@Tom, You can always split the memory space into two separate units. One half for server and one half for database. – Pacerier Oct 17 '14 at 20:31

Tom is correct on this. Some other reasons are that it isn't cost effective and that there are additional security risks.

Webservers have different hardware requirements than database servers. Database servers fare better with a lot of memory and a really fast disk array while web servers only require enough memory to cache files and frequent DB requests (depending on your setup). Regarding cost effectiveness, the two servers won't necessarily be less expensive, however performance/cost ratio should be higher since you don't have to different applications competing for resources. For this reason, you're probably going to have to spend a lot more for one server which caters to both and offers equivalent performance to 2 specialized ones.

The security concern is that if the single machine is compromised, both webserver and database are vulnerable. With two servers, you have some breathing room as the 2nd server will still be secure (for a while at least).

Also, there are some scalability benefits since you may only have to maintain a few database servers that are used by a bunch of different web applications. This way you have less work to do applying upgrades or patches and doing performance tuning. I believe that there are server management tools for making these tasks easier though (in the single machine case).

edited Oct 18 '11 at 15:05

answered Mar 18 '09 at 20:38

Dana the Sane
10.6k ● 7 ● 45 ● 73

1 If you're running anything but SPs then your webserver probably has full access to the data in your database anyways – George Mauer Mar 18 '09 at 20:42

Why isn't it cost efficient? Please specify. – Robert Jeppesen Mar 18 '09 at 20:57

Robert I expanded on that and added some comments on scalability and maintenance. – Dana the Sane Mar 18 '09 at 21:39

Security is a major concern. Ideally your database server should be sitting behind a firewall with only the ports required to perform data access opened. Your web application should be connecting to the database server with a SQL account that has just enough rights for the application to function and no more. For example you should remove rights that permit dropping of objects and most certainly you shouldn't be connecting using accounts such as 'sa'.

In the event that you lose the web server to a hijack (i.e. a full blown privilege escalation to administrator rights), the worst case scenario is that your application's database may be compromised but not the whole database server (as would be the case if the database server and web server were the same machine). If you've encrypted your database connection strings and the hacker isn't savvy enough to decrypt them then all you've lost is the web server.

answered Mar 18 '09 at 20:45

Kev
92.7k ●42 ●254 ●344

But you back up the database, right? Otherwise you'd be at just as much risk of losing it due to a hardware failure or rarely excited bug. An attack that kills the webserver will cause downtime all by itself. Enough privileges to add records to tables is enough to render a site useless. – Daniel Earwicker Mar 19 '09 at 0:04

Of course you backup the database, that's implicit, where in my post did I suggest otherwise. – Kev Mar 19 '09 at 0:10

1 Yes, but the conclusion therefore is that an attack intended to bring down the site can do so by destroying the web server config or the database, and the solution is the same for both: restore from backup. Specially protecting the database is (a) unnecessary and (b) impossible anyway. – Daniel Earwicker Mar 19 '09 at 0:38

@Earwicker - what about all the other databases residing on the DB server? All you've lost is one database. – Kev Mar 19 '09 at 0:55

6 Besides Daniel, you are missing a HUGE point. It's not about a database backup, it's about the compromised data. Would your customers be okay that "A hacker stole all your customer and sales data, but don't worry, I've got a backup." :) – Dylan - INNO Software Oct 9 '12 at 6:09

One factor that hasn't been mentioned yet is load balancing. If you start off thinking of the web server and the database as separate machines, you optimize for fewer network round trips and also it gets easier to add a second web server or a second database engine as needs increase.

answered Mar 18 '09 at 20:48

Paul Tomblin
130k ●44 ●273 ●365

I can speak from first hand experience that it is often a good idea to place the web server and database on different machines. If you have an application that is resource intensive, it can easily cause the CPU cycles on the machine to peak, essentially bringing the machine to a halt. However, if your application has limited use of the database, it would probably be no big deal to have them share a server.

answered Mar 18 '09 at 20:43

Mr. Will
1,991 ●2 ●16 ●26

I agree with Daniel Earwicker - the security question is pretty much flawed.

If you have a single box setup with a webserver and only the database for that webserver on it, if that webserver is compromised you lose both the webserver and only the database for that specific application.

This is exactly the same as what happens if you lose the webserver on a 2-server setup. You lose the web server, and just the database for that specific application.

The argument that 'the rest of the DB server's integrity is maintained' where you have a 2-server setup is irrelevant, because in the first scenario, every other database server relating to every other application (if there are any) remain unaffected as well - being, as they are, hosted elsewhere.

Similarly, to the question posed by Kev 'what about all the other databases residing on the DB server? All you've lost is one database.'

- if you were hosting an application and database on one server, you would only host databases on that server which related to that application. Therefore, you would not lose any additional databases in a single server setup when compared to a multiple server setup.

By contrast, in a 2 server setup, where the attacker had access to the Web Server, and by proxy, limited rights (in the best case scenario) to the database server, they could put the databases of every other application at risk by carrying out slow, memory intensive queries or maximising the available storage space on the database server. By separating the applications out into their own concerns, very much like virtualisation, you also isolate them for security purposes in a positive way.

answered Jan 11 '12 at 13:54

Oriental
105 ●2 ●7

It depends on the application and the purpose. When high availability and performance is not critical, it's not bad to not to separate the DB and web server. Especially considering the performance gains - if the appliation makes a large amount of database queries, a considerable amount of network load can be removed by keeping it all on the same system, keeping the response times low.

answered Mar 18 '09 at 20:37

simon

6,809 ● 18 ● 63 ● 98

Wow, No one brings up the fact that if you actually buy SQL server at 5k bucks, you might want to use it for more than your web application. If your using express, maybe you don't care. I see SQL servers run Databases for 20 to 30 applicaitions, so putting it on the webserver would not be smart.

Secondly, depends on whom the server is for. I do work for financial companies and the govt. So we use a crazy pain in the arse approach of using only sprocs and limiting ports from webserver to SQL. So if the web app gets hacked. The only thing the hacker can do is call sprocs as the user account on the webserver is locked down to only see/call sprocs on the DB. So now the hacker has to figure out how to get into the DB. If its on the web server well its kind of easy to get to.

answered Mar 18 '09 at 21:22

Jojo

189 ● 2 ● 10

+1, +1, +1 – Pacerier Oct 17 '14 at 20:34

I think its because the two machines usually would need to be optimized in different ways. Other than that I have no idea, we run all our applications with the server-database on the same machine - granted we're not public facing - but we've had no problems.

I can't imagine that too many people care about one machine being compromised over both since the web application will usually have nearly unrestricted access to at the very least the data if not the schema inside the database.

Interested in what others might say.

edited Mar 18 '09 at 20:39

answered Mar 18 '09 at 20:34

George Mauer

43.9k ● 99 ● 289 ● 504

1 George, I think you should refer to Mark Brackett's answer. The security your webserver has to the database would NOT be the same as it would be if the server was separate. Specifically the "local disk" wouldn't be accessible. In addition, if only a single website (of many) was hacked, they'd likely have only compromised access to THAT SITE's database (unless you use too powerful a user for that connection string). There are countless other scenarios, I just don't think a comment here is the place for them. – Dylan - INNO Software Oct 9 '12 at 6:07

I listened to that podcast, and it was amusing, but the security argument made no sense to me. If you've compromised server A, and that server can access data on server B, then you instantly have access to the data on server B.

answered Mar 18 '09 at 22:04

Daniel Earwicker

89.1k ● 31 ● 172 ● 251

Not true. You have access to whatever data or privileges that Box A had to Box B. In a secure setup, that would mean you have the highest level of DB access that the app on Box A has. You don't, however, have sa privs on the DB, or root on the OS of Box B. – Mark Brackett Mar 18 '09 at 23:49

2 "You have access to whatever data or privileges that Box A had to Box B" - that's what I meant by "you instantly have access to the data on server B". If the RDBMS was on box A and there was no box B, what difference would it make? NB. we're assuming you've already hacked box A, either way. – Daniel Earwicker Mar 19 '09 at 0:00

In a two box config, if you're applying the principle of least privilege, if box A (web) is compromised, then the worst that should have happened is you've lost the DB on box B and no more. – Kev Mar 19 '09 at 0:15

1 And on a one box config, if that one box is compromised, you've lost that one box, which includes the DB on it. What's the difference? – Daniel Earwicker Mar 19 '09 at 0:35

1 The difference is that on a two box config, if the web server is compromised, all you've lost is the web server and at worst the DB. The rest of the DB server's integrity is maintained. – Kev Mar 19 '09 at 1:20

Database licences are not cheep and are often charged per CPU, therefore by separating out your web-servers you can reduce the cost of your database licences.

E.g if you have 1 server doing both web and database that contains 8 CPUs you will have to pay for an 8 cpu licence. However if you have two servers each with 4 CPUs and runs the database on one server you will only have to pay for a 4 cpu licences

edited Oct 20 '09 at 13:27

answered Oct 20 '09 at 13:06

Ian Ringrose

28.9k ● 41 ● 187 ● 282

Please explain how this amounts to a savings. – John Saunders Oct 20 '09 at 13:08

1 John - he's saying that to get equivalent performance to 2 machines, you'd need to double the # of cores, which would double the SQL Server licensing costs. Why pay an extra $10-20k for SQL Server if all you get is the same performance as utilizing 2 machines. – Beep beep May 8 '11 at 3:00

only true if performance/resource utilization (e.g. cpu) is dominated by the app servers and not by the db server. if the db needs 8 cpus it won't work. – Andreas Dietrich Jan 9 at 12:14

An additional concern is that databases like to take up all the available memory and hold it in reserve for when it wants to use it. You can force it to limit the memory but this can considerably slow data access.

answered Mar 18 '09 at 20:47

HLGEM
75.6k ● 6 ● 81 ● 147

Arguing that there is a real performance gain to be had by running a database server on a web server is a flawed argument.

Since Database servers take query strings and return result sets, the data actually flowing from data server to web server is relatively small, but the horsepower required to process the query and generate the result set is relatively large. Optimizing performance around the data transfer time therefore is optimizing around the wrong thing.

Regarding security, there are advantages to having the data server on a different box than the web server. Having such a setup is not the be all and end all of security, but it is a step in the right direction.

Regarding scalability, it is easy and relatively cheap to add web servers and put them into cluster to handle increased traffic. It is not so easy and cheap to add data servers and cluster them. Also, web servers and data servers have different hardware needs, so multiple boxes help out with scalability.

If you are starting small and have only one box, then a good way would go would be to use virtual machines. Running the web server and data server in different VMs on one host gives you all the gains of separate boxes at the cost of one large box price.

edited Sep 19 '13 at 22:11

answered Sep 19 '13 at 22:03

Dave Anderson
1 ● 1

1 The original question asked about application servers, not necessarily public internet facing web servers. – João Bragança Nov 4 '13 at 18:26

@Dave, Huge strings can take alot of time..... – Pacerier Oct 17 '14 at 20:36

Operating system is another consideration. While your database may require larger memory spaces and therefore UNIX, your web server - or more specifically your app server since you mention only two tiers - may be a .Net-based, and therefore require Windows.

answered Mar 18 '09 at 21:58

Chris Noe
16.7k ● 22 ● 57 ● 88

I didn't down vote this, but "While your database may require larger memory spaces and therefore UNIX" - If you're running 64 bit windows and 64 bit SQL there's plenty of memory to play with. – Kev Mar 19 '09 at 13:48

Sorry, memory was meant as an example reason to need for deploying to split OSs. Other reasons include: performance, security, corporate standards/licensing, vendor support, etc. – Chris Noe Mar 19 '09 at 15:55

Ok! Here is the thing, it is more Secure to have your DB Server installed on another Machine and your Application on the Web Server. You then connect your application to the DB with a Web Link. Thanks it.

answered Mar 10 '13 at 23:41

Muroko
1

3 why it is more secure? – Bryan Chen Mar 10 '13 at 23:58