

Friday, December 1, 2017

### Why I Hate Commercial Control Panels

Whether it's DirectAdmin, cPanel, Plesk, or some other commercial hosting control panel, I pretty much dislike them all. They do their best to present an eye candy appealing image to help for good customer experiences, and I guess most of them do that well, probably far more appealing than the cobbled together scripts that I use allowing my hosts to manage their websites, but I guarantee you mine are probably more secure, efficient, scalable and redundant - capability beats looks any day!

In a few weeks time it'll be 2018, yet they still treat us like it's the 90's, limiting us to what daemons we can use on our own servers, limiting us to their choices of Linux distros, if it runs on RedHat, Debian, and FreeBSD, it'll run on SuSE, Slackware, Gentoo and others, but the biggest gripe of all, is they continue putting everything on one server. The process is along the lines of your customers sign up online, probably using Hostbill or WHMCS, it then shoots off to the current new hosts server's telling the Control Panel to set up the domain, its DNS, Email, Databases and Website structure - all on that one single machine, putting all your eggs in one basket, I guess they think hardware is infallible, where the truth is far from it.

The wheels do move ever so slowly though, back around 2010, some of these Control Panels started to introduce what they call multi server, or clustering - for DNS at least. How they seem to work though, means it still creates your DNS data on that one server, but then pushes out to the external DNS servers - which by the way for this to work, also needs your Control Panel installed, yep, more paid licences (some CP's do offer free DNS only licences and they are to be commended for that), either way, it's hardly an efficient method or true multi server, why bother creating records on the original host server in the first place, though still better than nothing I suppose, so when your host server is down for hardware failure, you should still at least have DNS.

But in web hosting, there are three critical components, DNS, WWW, and Email, so picture this, you're machines unresponsive, firefox and Outlook are both telling you it's DNS records can be found, but they just can't connect, what's the point if no-one can get to your website or email you, so your other two critical components, displaying your website or shopping cart etc, and Email, are kaput.

In 2017 there is no excuse for such archaic design, if your Website is down, your Mail shouldn't be, if Mail is down, your Website shouldn't be. Hosts, your customers wont care about blaming your Control Panel company, they will blame you, because it's your business and you decided how you are running it.

The situation doesn't have to be this bad though, but it will likely take making lots of noise to bring about change.

It's not hard, an ISP I worked for as far back as 2002, yes, 15 years ago, was able to do this right with complete separation of services, independent dedicated primary, secondary, and off-site tertiary DNS servers, we had multiple WWW servers which provided Web, MySQL and FTP for, at the time around 1200 customers each server, and we had a dedicated Mail server which stored all customers domains mail on RAID based disk array.

The way it should be done is very easy, instead of the Control Panel writing to corruptible flat, plain text files for Email virtual domains and users, a few simple SQL INSERT statements into a master database is all that's needed to have a highly scalable and efficient mail system, no need to edit text files for adding or closing domains, users, and suspension handling is as simple as setting a 0 or 1 in an "active" or "status" enum column.

I use the Postfix, Dovecot, MySQL (now days MariaDB) combination, I have used most MTA's over the years and found Postfix to be far superior in speed and resource niceness over Exim (which the CP companies tend to favour for some unknown reason), Qmail, and Sendmail, the great thing is Postfix and Dovecot work well together, using dovecot-lda for delivery is simple to configure, yes it does mean that if using multiple mail servers dovecot runs on all of them, even dedicated SMTP's (for delivery only) - but of course it only accepts connections for whatever the particular server is doing (truth be told every one of them can still do all protocols if you really wanted - some ISP's actually do it that way), and NFS to a NAS for the back end storage like EMC or NetApp, best using separate ethport on a private LAN, so mail storage flows go nowhere near the live network (but you could run it that way if you wanted to).

Separating Mail from Web also removes some of the loading issues that can be experienced during peak periods, it's

## Blog Export: Noel's Muses, <http://blog.ausics.net/>

not so much the mail delivery itself, it's all the necessary stuff like anti virus and anti spam checks in particular that induce load, couple that with a bunch of heavy poorly written CGI scripts and a lot of very active sites, and the machines response times become at times noticeably lagged, and that's not what you, your customers, or your customers customers want to see.

You can see from the above diagram how simple this is, your HostBill/WHMCS Server sets up customer for billing, then talks to the allocated customer Web Server Control Panel via an API which adds your domain/host etc on that server, it also fires off to external DNS servers to add records and then it creates domain/user entries in your master Mail DB located back on your Billing Server which in turn replicates out to the mail server(s).

Depending upon your size and mail flow, you may need multiple SMTP, POP3 servers (like above image), you could do it all on one Mail server - at least you've separated it from the Websites serving, or one server for each protocol, whatever you like, but redundancy is cheap in the long run, having multiple servers for each protocol, and behind a load balancer though just gives better resiliency and performance, and allows you to perform maintenance without affecting users.

Using a database makes it efficient, it allows multiple servers to access the same auth details, it's fast, especially with writes, and mysqldump's ease of backing them up, which I do and recommend hourly. I've done it that way for over ten years never having had a single corruption from MySQL or MariaDB and never needing a database backup, unlike the many times I and others have had to restore plain passwd files that were periodically only half written, even the old vpopmail cdb files were a problem at times, fast reads, but terrible with writes.

Does this all cost more? Of course it does, you're using more hardware, but it doesn't have to be the latest and greatest, eBay often has decent cheap server and networking hardware as do many refurbished tech dealers, and lets not forget that Google's Gmail runs on many cheap "home-type" PC boards, in much the same sort of design.

Local media has over the years covered many outages from providers who just stacked everything together with major SAN failures, although not to scale, the end result here is the same, all those Web, Mail, DNS services unreachable, one provider I recall had over 10K angry customers without any service for two days with their SAN failure. Sometimes you just have to care enough to outlay a little bit more for long term benefit.

So if we on a shoe string budget could do this way back in 2002, why, in almost 2018, do the mass produced commercial Control Panel companies treat us all like it's still the 90's and make life more difficult than it needs to be for Host providers.

Well cPanel, Plesk, DA and co? Balls in your court, it's time you got your arses into gear and bring your software into the modern era. I'm sure there's a buck to be made for a true mutli-server operation, do I hear you say new product? I certainly hope so.

Posted by NoelB at 09:11