

Strategy: Diagonal Scaling - Don't Forget to Scale Out AND Up

Monday, November 5, 2007 at 2:35PM

Todd Hoff in [Diagonal Scaling](#), [Horizontal Scaling](#), [Strategy](#), [Vertical Scaling](#)

All the cool kids advocate scaling out as the secret sauce of scaling. And it is, but don't forget to serve some tasty "scaling up" as a side dish. Scaling up doesn't have to mean buying a jet propelled, liquid cooled, 128 core monster super computer. Scaling up can just mean buying at the high end of the commodity buffet by buying more cores, more memory and using a shared nothing architecture to take advantage of all that power without adding complexity. Scale out when you need to, but big beefy boxes can absorb a lot of load before it's necessary to hit up your data center for more rack space. Here are a few examples of scaling out and up:

[John Allspaw](#), Flickr's operations manager, coined the term *diagonal scaling* for this strategy. In [Making a site faster by removing machines](#) (and a comment on this post) John told how Flickr replaced 67 dual-cpu boxes with 18 dual quad-core machines and recovered almost 4x rack space and reduced costs by about 50 percent.

[Fotolog's strategy](#) is to scale up and out. By adding more cache, more RAM, more CPUs, and more efficient CPUs they were able to handle many millions more users with the same number of machines. This was a conscious choice on their part and it worked beautifully.

[Wikimedia](#) says scaling out doesn't require using cheap hardware. Wikipedia's database servers these days are 16GB dual or quad core boxes with 6 15,000 RPM SCSI drives in a RAID 0 setup.

Kevin Burton in his [Distributed Computing Fallacy #9](#) post also says scaling out doesn't mean cheap:

We're seeing machines with eight cores and 32G of memory. If we were to buy eight disks for these boxes it's really like buying 8 machines with 4G each and one disk. This partially goes into the horizontal vs vertical scale discussion. Is it better to buy one \$10k box or 10 \$1k boxes? I think it's neither. Buy 4 \$2.5k boxes. The new multicore stuff is super cheap.

Jeremy Cole in [Scaling out AND up, a compromise](#) asks for compromise:

Scaling out doesn't mean using crappy hardware. I think people take the "scale out" model (that they've often only read about from outdated conference presentations) to quite an extreme. They think scaling out means using desktop-class, bad hardware, and just buying a ton of them. That model doesn't work, and it's hell to maintain in the long term.

Use commodity hardware. You often hear the term "commodity hardware" in reference to scale out. While crappy hardware is also commodity, what this means is that instead of getting stuck on the low-end \$40k machine, with thoughts of upgrading to the \$250k machine, and maybe later the \$1M machine, you use

data partitioning and any number of let's say \$5k machines. That doesn't mean a \$1k single-disk crappy machine as said above. What does it mean for the machine to be "commodity"? It means that the components are standardized, common, and the price is set by the market, not by a single corporation. Use commodity machines configured with a good balance of price vs. performance.

Article originally appeared on High Scalability (<http://highscalability.com/>).

See website for complete article licensing information.