



# Database Virtualization and the Cloud

*How database virtualization, cloud computing and other advances will reshape the database landscape*

by  
Mike Hogan, CEO  
ScaleDB Inc.

December 10, 2009

## Introduction

In nature, small moves in tectonic plates cause pressure to build over time until it is released in sudden and violent earthquakes. The same principle applies to the world of computer technology. Technical advances often build upon each other, paving the way for large disruptive shifts in the technology landscape. For example, the combination of the Internet, browsers and AJAX/Flash enable online applications like Google Docs to threaten desktop applications.

We are now poised for just such a disruption in the database market. The technologies that are reshaping the database landscape are: faster storage, faster networking/interconnections, virtualization and cloud economics. This white paper explains each of these technologies and how they are aligning to create a tectonic disruption in the database landscape.

Designing a database management system (DBMS) is an exercise in identifying and minimizing the various performance constraints imposed by the current computing technologies. In the past, two of the biggest performance bottlenecks have been shared storage devices and the network connecting those storage devices to the database. The combination of shared storage and slow network connections introduced severe performance bottlenecks—in addition to higher costs—putting the shared-disk DBMS architecture at a significant disadvantage to shared-nothing. This explains why the shared-nothing DBMS became the de facto standard.

This paper describes how changes in the underlying technologies, combined with the economics of cloud computing, reframe the shared-disk versus shared-nothing debate. Technological advances have put shared-disk performance on par with shared-nothing, while cloud computing and virtualization strongly favor the shared-disk architecture. These changes are poised to dramatically alter the database landscape, particularly for cloud databases.

## A Primer on Shared-Nothing and Shared-Disk Databases

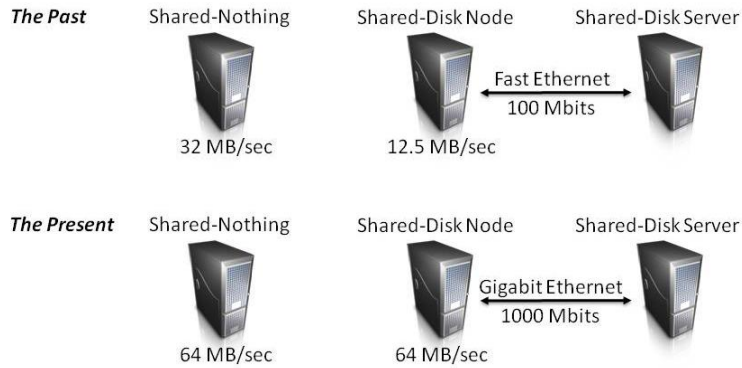
The two primary DBMS architectures are shared-nothing and shared-disk. Shared-nothing databases split or partition the data so that each database server exclusively processes and maintains its own piece of the database. Shared-disk is analogous to a single large trough of data, where any number of database nodes can process any portion of that data. Based upon traditional computing constraints, the shared-nothing architecture has been the price-performance leader. That is now changing.

## The Technology Drivers

### Interconnect Performance:

Network connections have historically acted as a throttle on data flow. In the not too distant past, a standard Ethernet connection (10 Mbits) and even a Fast Ethernet connection (100 Mbits) would significantly throttle data flow. The disk itself might deliver 32 MB/s, but a 10 Mbit Ethernet connection would limit the data to each node to 1.25 MB/s and a 100 Mbit Fast Ethernet connection only delivers 12.5 MB/s. The math is simple; a local disk drive delivers data 2.56-times faster than Fast Ethernet and 25.6-times faster than standard Ethernet.

**Diagram 1 - Interconnect Speeds: Shared-Nothing vs. Shared-Disk**



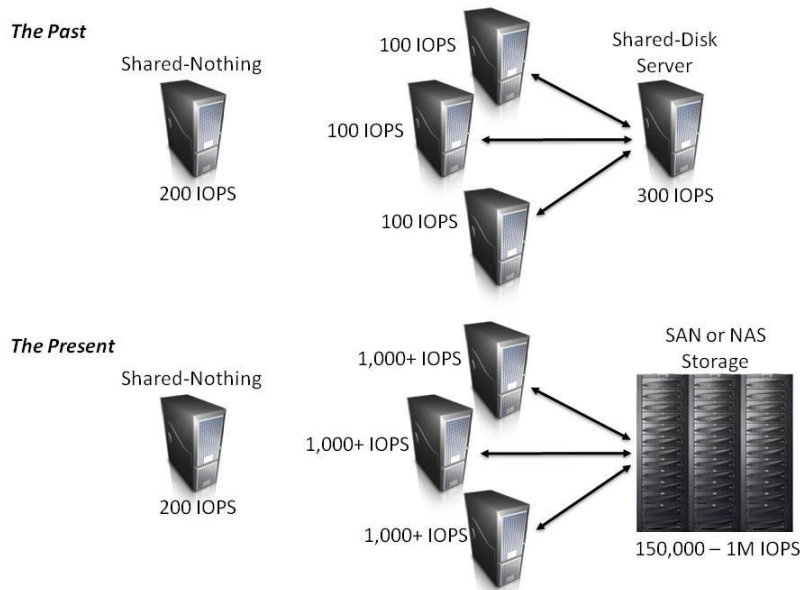
Shared-nothing databases use the local disk, while shared-disk databases rely on storage that is shared and accessible via the network. Because of this, shared-nothing databases had an inherent performance advantage in an era of slow networking. Today, Gigabit over Ethernet (GBoE) is standard, and we have faster interconnections like Infiniband (up to 12 GB/s) and Fiber Channel (5.1 GB/s) for even better performance.

Improvements in networking (interconnect) performance have eliminated the performance shared-nothing DBMS have always enjoyed. In other words, with regard to interconnect performance, shared-disk DBMS are every bit as fast as shared-nothing DBMS.

**Storage Performance:**

CPUs are faster than storage. The drive head on a disk is limited by physics because it has to move to various sectors on the disk to read the information. Since a single disk is slower than a CPU, it makes no sense to then share that slow disk across multiple CPUs. Even if you used an expensive disk that gave you a 50% increase in performance—for a significant additional cost—sharing that disk across multiple CPUs would just make the performance bottleneck even more problematic.

**Diagram 2 - Storage Speeds: Shared-Nothing vs. Shared-Disk**



A variety of technologies including tiered storage (exploiting faster storage media like battery-backed RAM, SSD, etc.) and stripping the data across multiple disks—to eliminate the constraints imposed by drive head movement—have resulted in dramatically faster storage. Modern storage devices can deliver 150,000 to 1,000,000 IOPS. Even if you share this across multiple data nodes, you still get far better performance than you would from a local disk, which delivers about 200 IOPS. In fact, Amazon’s Elastic Block Services (EBS)—a sort of SAN in the cloud—delivers 1,000 IOPS per instance. Stripe your data across five instances of EBS and you get about 5,000 IOPS, which is 25X the performance of local storage. In short, shared storage is no longer a performance bottleneck; it is actually a performance accelerator.

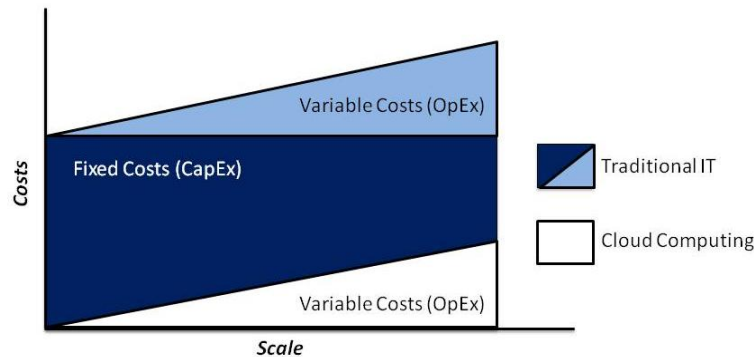
**Storage Costs:**

Based upon the advances described above, it is clear that using shared storage can actually improve database performance, but at what cost? Previously, the shared-nothing database architecture has enjoyed advantages in both performance and cost. As described above, it no longer enjoys a performance benefit. Now we will address the cost issue.

Shared storage devices such as NAS and SAN are typically very expensive. However, cloud economics turn the storage cost equation on its head. Amazon’s relational database services (RDS) costs 11 cents per hour or \$79.00 per month<sup>1</sup>, while Amazon EBS is priced on usage, but the example Amazon provides is \$36.00 per month<sup>2</sup> for a 100GB database. In other words, storage is about half the price of a computer instance. This is radically different from traditional pricing where a server might cost a couple thousand dollars, but storage could cost \$100,000 or 50-times the cost of a single server.

Cloud computing is disrupting traditional database economics. By leveraging multi-tenancy—the ability to share a storage infrastructure across multiple users—the cost of a large-scale storage infrastructure can be amortized across multiple companies. This replaces the fixed costs (Capital Expenditures) and variable operational costs (Operational Expenditures) inherent in an in-house data center, with a variable cost-only economic model (see Diagram below).

*Diagram 3 – Cloud Computing Economics*



Ironically, the traditional cost advantage of the shared-nothing DBMS architecture is now a cost disadvantage. Shared-nothing implementations have avoided high-cost storage, but have then relied on slave servers using commodity hardware. The cloud makes high-performance storage less expensive

<sup>1</sup> <http://aws.amazon.com/rds/#pricing>

<sup>2</sup> <http://aws.amazon.com/ebs/>

than the additional compute instance required for a single slave. As a result, cloud economics make shared-disk DBMS less expensive than shared-nothing DBMS.

### Software Costs:

One of the primary advantages of the shared-disk architecture is the high-availability it delivers. Shared-disk databases are also easier to set-up and maintain, since the DBA doesn't have to worry about slaves, replication, promotion, etc. The flipside of this equation is that the shared-disk DBMS is very challenging for software developers to create. As a result, shared-disk DBMS are only available from a couple of companies and those companies charge a hefty premium for their software. ScaleDB is changing this pricing paradigm.

ScaleDB operates as a storage engine for MySQL. This enables ScaleDB to build upon a stable, fast and market-leading database. This significantly reduces ScaleDB's risks and costs. ScaleDB then passes on these savings to its users in the form of licensing costs that are pennies on the dollar relative to competing shared-disk options from Oracle and IBM. In fact, ScaleDB demonstrates a *lower* total cost of ownership (TCO) than free open source storage engines using the shared-nothing architecture.

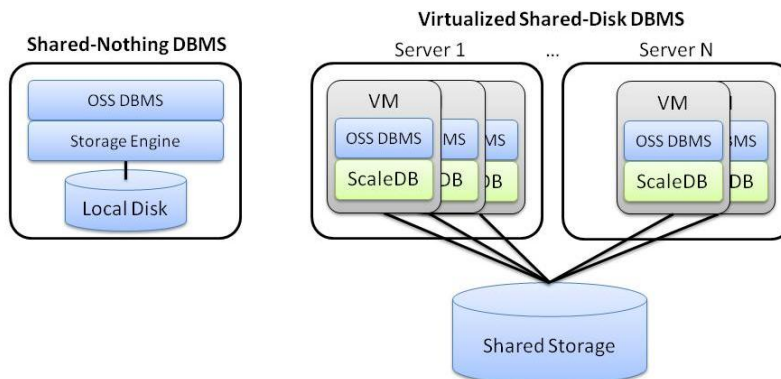
### Virtualization:

The primary driver of cloud computing's cost advantages is virtualization. Virtualization is the ability to create, operate and manage computing instances independent of the underlying hardware. Virtualization delivers many advantages but from the perspective of the cloud company, the greatest advantage is cost savings.

Prior to virtualization, you would dedicate specific servers to each application. Those servers would be sized to something in excess of the projected peak load of that application over some future time horizon. Virtualization enables you to start with the smallest instance and then scale elastically as needed. You only use what you need and you only pay for what you use.

The one fly in the ointment is that shared-nothing databases don't work in virtualized environments. When using a shared-nothing architecture, applications must be hardwired to specific database servers. Those databases are then hardwired to their specific data partitions, you can't virtualize them. So, while web servers, applications, middleware and storage can all exploit virtualization, databases require a dedicated server. Since that database server is static, you must size that server for your peak load. As a result, you'll get the cost advantages of cloud computing at all levels *except* the database level.

**Diagram 4 - Virtualization: Shared-Nothing vs. Shared-Disk**



The shared-disk DBMS relies on a cluster of identical database nodes processing a single trough of data. These identical database compute nodes are ideal for virtualization. As a result, you can scale your database computing elastically on demand, so you only pay for what you need when you need it. It does away with dedicated and underutilized servers. This enables the cloud hosting company to better utilize their infrastructure, saving money for both you and them.

Let's look more closely at the cost differential between using a large database instance and a scalable shared-disk DBMS like ScaleDB, where you only pay for what you use. The benefit of this model is evidenced by the cost differential for Amazon RDS, where the cost of a single instance can grow from 11¢ per hour to \$3.10 per hour<sup>3</sup>.

The largest Amazon RDS Instance: .....	\$3.10/hour
Average Utilization.....	20%
Cost of a dynamically scalable DBMS (pay for what you use) .....	\$.65

*Assumptions:* You typically allocate at least 20% of the total server capacity for future growth, so you are working with only 80% of the server's capacity. Of that remaining 80%, you might peak for say an hour a day, while the bulk of the time it is utilizing very little of the server's capacity. If you average out the utilization, it is pretty typical that the average utilization falls below 20%, which would mean that a pricing model that only charges you for what you use, would cost \$.65 or 1/5<sup>th</sup> the cost of the large server. This is precisely what the shared-disk DBMS does, it only runs the servers you need, when you need them.

Adding a slave server for failure doubles the server cost, bringing it to \$6.20/hour. Note: RDS doesn't support slaves, but you could run a slave in your own AMI. If you are running a shared-nothing DBMS with a slave server and paying \$6.20 per hour, your server costs are 10X what they would be if you were using a shared-disk DBMS. This is yet another example of how shared-disk is a more efficient DBMS architecture for the cloud.

## Conclusion

The shared-nothing DBMS architecture gained widespread adoption on the basis of performance and cost advantages that no longer exist. Shared storage, with the help of extremely fast interconnects, now delivers data to the CPU several times faster than a local disk. Cloud computing economics—leveraging the power of multi-tenancy—delivers extremely fast shared storage at a dramatically reduced cost. Virtualization then compounds these advantages by enabling users to scale elastically and to pay only for the resources they use. The cost/performance advantages have decisively shifted in favor of the shared-disk DBMS. It is just a matter of time before the shared-disk DBMS establishes dominance in the cloud. I suspect we will also see some significant disruption in the database landscape for enterprises and private clouds as well.

---

<sup>3</sup> <http://aws.amazon.com/rds/#pricing>