

# Sharding: In Theory and Practice (Part Five)

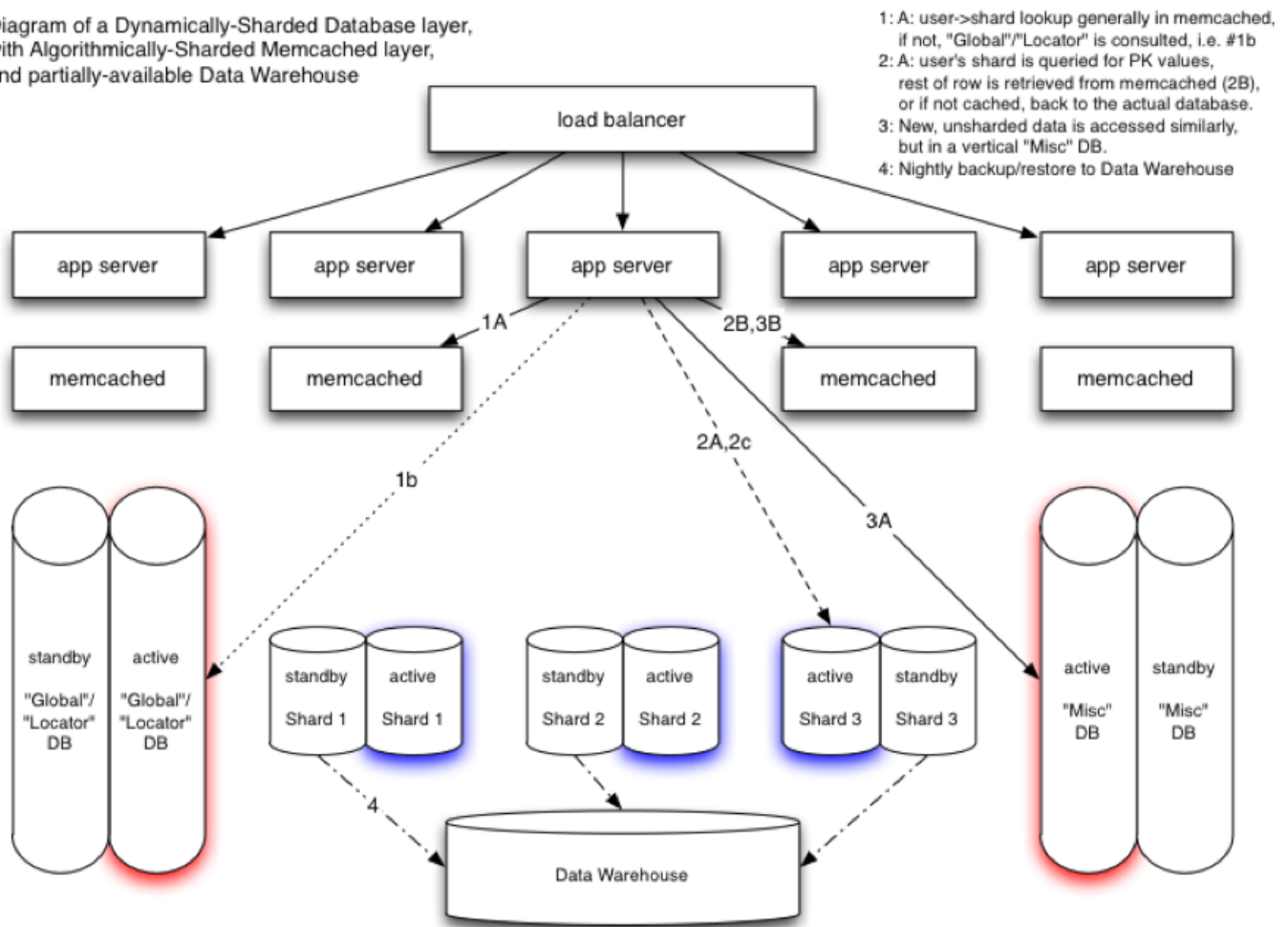
📌 *Standard* / 👤 *by Neil Harkins (https://www.clustrix.com/author/neil-harkins/)* / 📅 *March 6, 2013* / 💬 *No Comments (https://www.clustrix.com/bettersql/sharding-theory-practice-part-five/#respond)*

## Part Five: The Data Warehouse

Welcome to the final installment of this series on database sharding. Following the steps in part two (The Differences Between Algorithmic and Dynamic Sharding) and part three (What's in a Shard?) of this series, you can achieve a scalable and fault-tolerant architecture, but one that sacrifices many features of a relational database and is quite costly to maintain.

See the illustration below for the complete picture:

Diagram of a Dynamically-Sharded Database layer, with Algorithmically-Sharded Memcached layer, and partially-available Data Warehouse



Although this architecture can provide service to millions of customers, those customers simply won't stick around unless the service can also quickly add new features to stay relevant. The hope is that several of your engineers can now move from sharding infrastructures to implementing features that are noticeable to the customer base. At least, more noticeable than a decrease in the frequency of an unexpected downtime page, like Twitter's iconic "fail whale." It should also be noted, however, that sharding often creates more and varied failure modes, corresponding to the additional moving parts that were introduced.

It's important to understand how customers use your service in order to determine the direction of new features, and even create new revenue opportunities from existing features. This is the promise of Big Data and data science – but the answers to those questions are now split across multiple shards.

In some cases, aggregate functions can be performed on each shard, and then again on the superset of those results, which is the simplest example of MapReduce. But there are many powerful reporting tools that are simply not shard-aware. Since everyone shards a little differently, there isn't a reasonable number of targets for reporting software companies to develop against.

My previous employer, SixApart, chose Mondrian by Pentaho software for its reporting and billing needs because it supports MySQL as a backend. But to use this tool, every shard had to be combined into the single database that Mondrian expects. Luckily, the need for sharding at SixApart was due to request load, not size, so the largest table could physically fit in the largest innodb file that a single mysql instance could support (but that was yet another wall we were bound to hit at some point in the future).

## The Care and Feeding of a RAID 00 Beast

The data warehouse implemented by SixApart was a single large machine with a RAID 00 storage subsystem – basically software striping on top of hardware striping for faster writing of data.

Every evening, cron executed mysqldump on each shard in parallel (the passive side, to be precise). Then, the large storage volume was wiped and each shard's backup was restored onto it. This process took about 14 hours every day, leaving only ten hours for the analytics team to use the data warehouse. Large OLAP queries can run for minutes to hours, which makes that small, four-hour window a significant limitation. Also, dumps or restores often failed and left us completely unable to generate reports that day.

We completely wiped the data warehouse because MySQL has no incremental backup and because of the single-master limitation in MySQL's replication.

Clustrix has no such limitation. Basically, any number of slave processes can be defined, each with a different master, e.g. each shard. This functionality has sometimes been called “Fan-In” replication. This configuration can provide a data warehouse with 24/7 uptime, i.e. a “real-time” rather than a day old snapshot, and no size limitations looming on the horizon (I quote “real-time” because MySQL replication is still asynchronous, so it's usually a second or more behind the master).

A few years ago, SixApart merged with VideoEgg to form a new company called SayMedia. I asked if they were still casino using the RAID 00 Beast, but they inherited a Hadoop cluster in the merger and are trying to use it for their data warehousing needs. I wished them luck.

## The Mousetrap

During one interview at SixApart, an applicant was asked, “How does the Internet work?” She jokingly replied, “There are a bunch of hamsters running in those wheels.” I laughed, but also saw some truth in it – members of our Ops department staff were like the hamsters constantly keeping things running.

Despite the diagram above resembling something created by Rube Goldberg or Dr. Seuss, it was implemented by some bright people using the only materials they had available at the time – materials simply not designed for scalability.

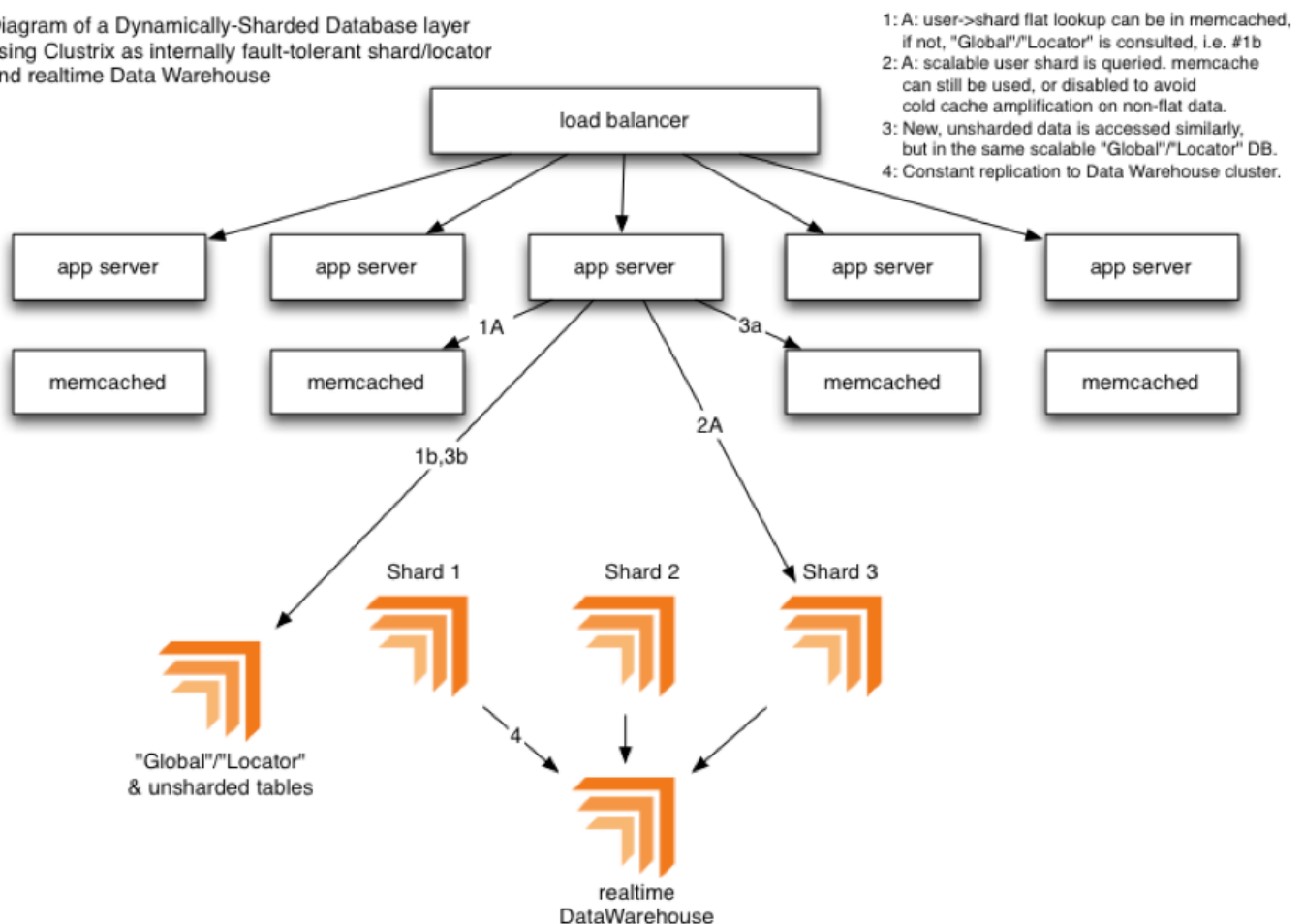
Like a modern day Tower of Babel, the technical community itself has sharded into factions around each implementation. All of these factions doing the same things using different dialects, with experts expected to be experienced with all of them.

Think of how many advances could have been made by mid-size Internet companies if they didn't sink so much time and money into designing for scale. I'll even assert that Google's many advances have been possible because they recognized problems common to multiple teams and solved them once instead of each team reinventing the wheel.

## Refactor your Architecture

It's never too late to improve your sharding architecture by replacing problem-wrought components with a building block designed for scale and fault tolerance. Because we're a drop-in replacement for MySQL, your application and sharding logic does not need to change (see new diagram below).

Diagram of a Dynamically-Sharded Database layer using Clustrix as internally fault-tolerant shard/locator and realtime Data Warehouse



Had Clustrix started a few years earlier, I don't think sharding would be as prevalent as it is today. But alas, relational databases have complex functionality (that allows for less complex applications that can be deployed much faster), so most DevOps choose to shard instead of contributing code to MySQL.

It takes years for a small team of engineers to write a database from the ground up with scalability and fault tolerance in mind. At Clustrix, we tackled that hard problem for you. In comparison, Oracle had decades and enormous resources to address the problem, and yet still, they didn't.




**Part One: A Brief History of Sharding** (<https://www.clustrix.com/bettersql/sharding-theory-practice-part-one/>)


**Part Two: The Differences Between Algorithmic and Dynamic Sharding**  
(<https://www.clustrix.com/bettersql/sharding-theory-practice-part-two/>)

**Part Three: What's in a Shard?** (<https://www.clustrix.com/bettersql/sharding-theory-practice-part-three/>)

**Part Four: Using Memcached** (<https://www.clustrix.com/bettersql/sharding-theory-practice-part-four/>)

**Part Five: The Data Warehouse** (<https://www.clustrix.com/bettersql/sharding-theory-practice-part-five/>)

 (<https://www.facebook.com/Clustrix>)  (<https://twitter.com/Clustrix>) 

(<https://plus.google.com/111948679378130082453>)  (<https://www.linkedin.com/company/clustrix>)

## Product

Overview  
(<https://www.clustrix.com/summary-of-our-db/>)  
Cloud Database  
(<https://www.clustrix.com/cloud-database/>)  
Elastic Scale  
(<https://www.clustrix.com/elastic-scale/>)

## Case Studies

Hit Labs  
(<https://www.clustrix.com/resources/customer-success-story-hitlabs/>)  
Match.com  
(<https://www.clustrix.com/resources/customer-success-story-two-com/>)

## Recent News

MySQL-compatible  
cloud database cost  
comparison  
(<https://www.clustrix.com/bettersql/mysql-compatible-cloud-database-cost-comparison/>)  
Ad Tech Carousel  
(<https://www.clustrix.com/slideshow-ad-tech-carousel/>)

## Support

(<https://support.clustrix.com/>)  
Documentation  
(<http://docs.clustrix.com>)  
Blog (/bettersql/)  
Resources  
(<https://www.clustrix.com/resources/>)  
Free Trial  
(<https://www.clustrix.com>)