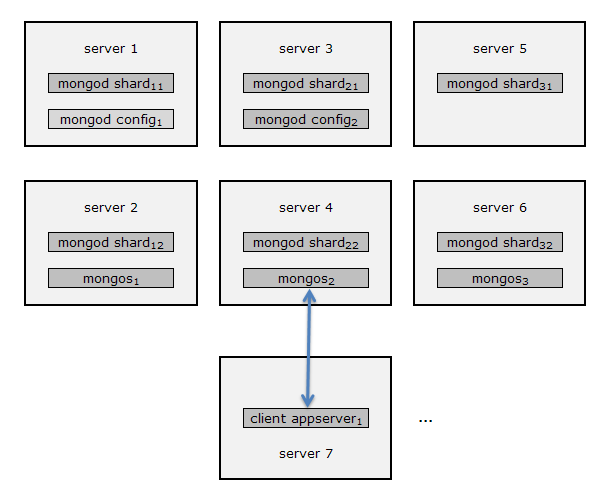
MongoDB 集群中包含一个自动分片模块 (“[mongos](http://blog.nosqlfan.com/tags/mongos)”). 自动分片可以用于构建一个大规模的可扩展的数据库集群,这个集群可以并入动态增加的机器。自动建立一个水平扩展的数据库集群系统，将数据库分表存储在[sharding](http://blog.nosqlfan.com/tags/sharding" \o "查看 sharding 的全部文章" \t "_blank)的各个节点上。在一个mongodb的集群中包括一些shards(mongod进程)，mongos的路由进程，一个或多个config服务器。sharding是一种对大规模数据存储的一种策略，关于sharding的详细信息可以查看[这里](http://en.wikipedia.org/wiki/Shard" \t "_blank)。也许有人会问，为什么需要做这种策略，因为在一个大型系统中最后的瓶颈会落在网络的带宽和磁盘的读写上，如果将数据分布在多个机器上的多个磁盘上，将会系统数据的处理能有所提高。

**MongoDB 集群的结构**：  
下图中Shard是指每个节点的shard有一个或更多的服务器和存储数据的mongod进程，而mongod是MongoDB数据的核心进程。

每台机器上的mongod从配置获取服务器(元数据[metadata](http://blog.nosqlfan.com/tags/metadata" \o "查看 metadata 的全部文章" \t "_blank) )，然后，当收到客户端请求时，它请求路由到相应的服务器组和编译结果发送回客户端。

mongos进行可以被看作是一个路由和协调的过程，因为他可以使得每个单一的各个节点组成一个集群系统。 另外还需要强调一点mongos进程没有持久状态，每个实例都需要一定的数据存储的内存空间。

换而言之，所谓MongoDB 集群也就是 MongoDB做了一个数据库路由的策略，而且保证跨库操作的数据库事务，而MongoDB 集群中的关键部分Sharding不是一门新技术，而是一种策略，关键还是看应用场景和案例提供的可用性，因为Sharding不仅仅是MongoDB 集群中所提到的分布在不同的机器上，还可以分表，分区，分数据，等等。



**MongoDB 集群的工作原理：**

其中有一个服务器上存储着集群的metadata信息，包括每个服务器，每个shard的基本信息和chunk信息Config Server 主要存储的是chunk信息。每一个config服务器都复制了完整的chunk信息，就是下图中左边黄色的部分。

如果客户端对集群的MongoDB插入一条数据，客户端并不知道刚刚插入的数据被分配到具体哪个MongoDB节点上了，因为当一条数据被传入 MongoDB集群中通过mongos路由，所以我们并感觉不到是数据存放在哪个shard的 chunk上，但是通过后台的Sharding的管理命令可以看到插入的数据存放在哪个节点上。

