

# MongoDB聚合计算性能优化研究

肖子达<sup>1,2</sup> 朱立谷<sup>1,2</sup> 冯东煜<sup>1,2</sup> 张迪<sup>1,2</sup>

<sup>1</sup>(中国传媒大学计算机学院 北京 100024)

<sup>2</sup>(安防大数据处理与应用北京市重点实验室 北京 100094)  
(shawzida@gmail.com)

在大数据应用方面，基于 NoSQL 的 OLAP 提供基本的聚合分析功能。MongoDB 作为 NoSQL 数据库中的一类，其内置聚合功能使用 MapReduce 的计算框架是基于两个方面的考虑。一方面，数据库可以通过扩展集群能够显著提高聚合计算的性能。而另一方面，在有限的物理资源下，结合业务的数据库配置方法可以提高计算性能。

方法的第一步是，估计读写操作的工作负荷并选择合适的片键。一个好的片键应该有足够的基数粒度，并能达到写扩展和查询分离的目的。

方法的第二步是，选择合适的片键类型。片键类型确定数据在集群中的分布方式。片键的类型分为三种：连续型、哈希型、标签型。默认连续型和标签型都是比较适用于 MapReduce 的运行环境。

方法的最后一步是消除相关的索引。当查询需要返回集合中的 2%~60% 时就应该对索引查询和全表查询进行比较。创建片键后，MongoDB 可以手动删除该索引，使得在进行 MapReduce 时采用全表查询的方式。而我们的建议是对于需要聚合操作的数据集合，不对作为片键的字段进行索引。

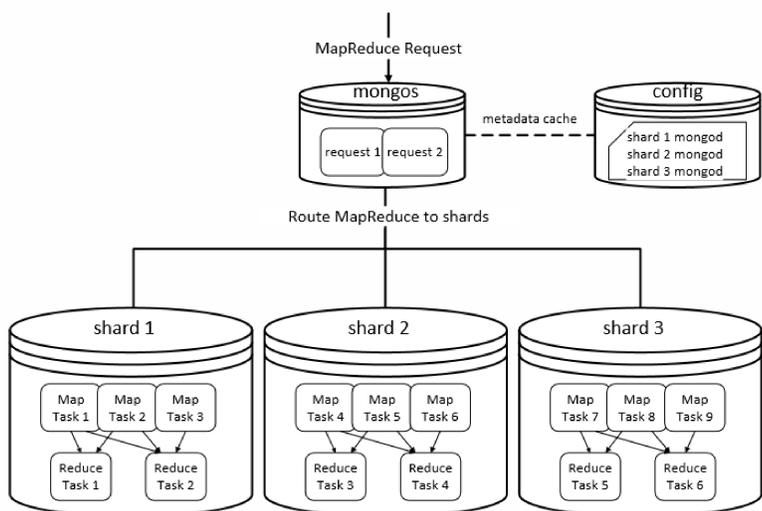


图1 MongoDB的MapReduce模型

在分片集群的情况下，MongoDB 还能进行 MapReduce 并行计算。图 1 展示的 MapReduce 在分片集群下的并行计算。作为 MongoDB 的路由节点，mongos 将客户端发出的请求进行暂存，并询问存储各分片信息的 config 节点，以获得相关数据的位置。mongos 得到数据的分布后，将请求发给各个分片节点 shard。各 shard 节点进行 mapper 和 reducer 处理，并将处理数据返回给 mongos。mongos 将处理完的数据进行合并以及最后的 reducer。

MongoDB 的聚合操作是一种消耗内存资源和磁盘读性能的计算过程。MongoDB 的 MapReduce 是最灵活的计算接口。通过建立分片集群，可以有效提高 MapReduce 的运算能力。在已有硬件资源一定条件下，我们可以采取一种基于数据特征的优化方法来提升现有集群的聚合运算能力。



图2 优化方法流程图

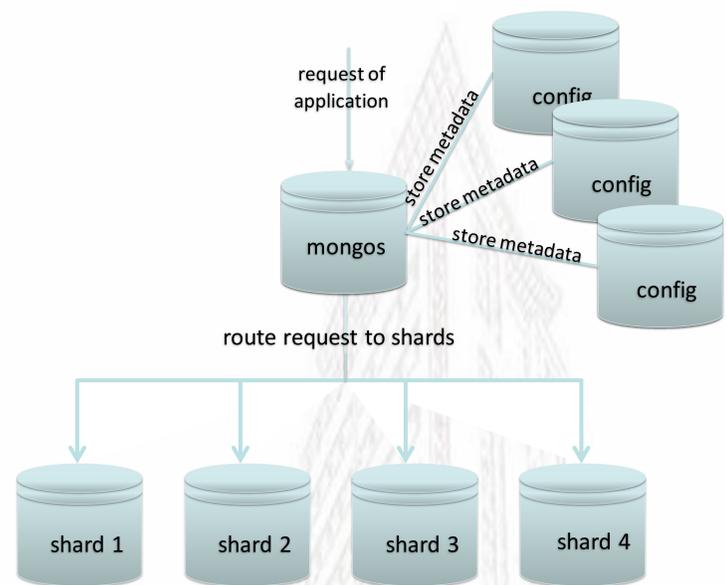


图3 分片集群实验环境

实验设计的目的是为了探索不同片键对于聚合计算的影响并得出使用我们提出的方法选择的环境是否对聚合计算最优。针对不同的应用场景，片键的均匀性、随机性、局部性特征对插入与读取操作的影响各不相同。实验的分片集群如图 3。MongoDB 采用 8 节点分片集群，其成员配置如下：4 个分片节点、3 个配置节点、1 个路由节点。

通过实验我们可以得出，采用高基数粒度的分片片键能够让数据在节点上分布更加均匀。当聚合计算的查询字段是低基数粒度的字段时，可以与高基数粒度组合作为复合片键来满足数据计算和存储。而另一种片键选择的方法适用于部分业务场景，这时片键应该在较高粒度的字段中选择最能提高聚合计算速度的字段，而不应满足数据的均匀分布。并且，我们认为将历史统计分析和实时查询的数据存储隔离的方法，能避免这种无索引即席查询性能的降低。