- to ensure that mongos can isolate most queries to a specific mongod.

Furthermore:

- Each shard should be a *replica set*, if a specific mongod instance fails, the replica set members will elect another to be *primary* and continue operation. However, if an entire shard is unreachable or fails for some reason, that data will be unavailable.
- If the shard key allows the mongos to isolate most operations to a single shard, then the failure of a single shard will only render *some* data unavailable.
- If your shard key distributes data required for every operation throughout the cluster, then the failure of the entire shard will render the entire cluster unavailable.

In essence, this concern for reliability simply underscores the importance of choosing a shard key that isolates query operations to a single shard.

Sharded Cluster Query Routing

MongoDB mongos instances route queries and write operations to *shards* in a sharded cluster. mongos provide the only interface to a sharded cluster from the perspective of applications. Applications never connect or communicate directly with the shards.

The mongos tracks what data is on which shard by caching the metadata from the *config servers* (page 670). The mongos uses the metadata to route operations from applications and clients to the mongod instances. A mongos has no *persistent* state and consumes minimal system resources.

The most common practice is to run mongos instances on the same systems as your application servers, but you can maintain mongos instances on the shards or on other dedicated resources.

Note: Changed in version 2.1.

Some aggregation operations using the aggregate command (i.e. db.collection.aggregate()) will cause mongos instances to require more CPU resources than in previous versions. This modified performance profile may dictate alternate architecture decisions if you use the *aggregation framework* extensively in a sharded environment.

Routing Process

A mongos instance uses the following processes to route queries and return results.

How mongos Determines which Shards Receive a Query A mongos instance routes a query to a *cluster* by:

1. Determining the list of *shards* that must receive the query.
2. Establishing a cursor on all targeted shards.

In some cases, when the *shard key* or a prefix of the shard key is a part of the query, the mongos can route the query to a subset of the shards. Otherwise, the mongos must direct the query to *all* shards that hold documents for that collection.

Example

Given the following shard key:

```
{ zipcode: 1, u_id: 1, c_date: 1 }
```

Depending on the distribution of chunks in the cluster, the mongos may be able to target the query at a subset of shards, if the query contains the following fields: