Assume you have a MongoDB collection which ocupies 6 chunks **evenly distributed** in 3 shards (i.e., 2 chunks per shard). Being the document ID also the shard key, the chunk of a document is determined **by means of a hash function**. Assuming that accessing one document takes one time unit (existing indexes are used at no cost) and we have 6.000 documents in the collection, *k* of which have value "YYY" for attribute "other", how many time units would take the following operations:

a) *FindOne({ _id: "XXX" })*

b) *Find({_id: { $in: [1,..,3000] })*, being [1,..,6000] the range of existing IDs.

c) *Find({ other: "YYY" })*, being the attribute indexed.

d) *Find({ other: "YYY" })*, being the attribute NOT indexed.

Note: As typically in RDBMS optimizers, assume uniform distribution of values and statistical independence between pairs of attributes.

Assume you have a MongoDB collection which ocupies 6 chunks **UNevenly distributed** in 3 shards (i.e., 1, 2 and 3 chunks per shard respectively). Being the document Id also the shard key, the chunk of a document is determined **by means of a hash function**. Assuming that accessing one document takes one time unit (existing indexes are used at no cost) and we have 6.000 documents in the collection, *k* of which have value "YYY" for attribute "other", how many time units would take the following operations:

e) FindOne({ _id: "XXX" })

f) *Find({_id: { $in: [1,..,3000] })*, being [1,..,6000] the range of existing IDs.

g) *Find({ other: "YYY" })*, being the attribute indexed.

h) *Find({ other: "YYY" })*, being the attribute NOT indexed.

Note: As typically in RDBMS optimizers, assume uniform distribution of values and statistical independence between pairs of attributes.

Assume you have a MongoDB collection which ocupies 6 chunks and is **evenly distributed** in 3 shards (i.e., 2 chunks per shard) . Being the document Id also the shard key, the chunk of a document is determined **by range**. Assuming that accessing one document takes one time unit (existing indexes are used at no cost) and we have 6.000 documents in the collection, *k* of which have value "YYY" for attribute "other", how many time units would take the following operations:

i) FindOne({ _id: "XXX" })

j) *Find({_id: { $in: [1,..,3000] })*, being [1,..,6000] the range of existing IDs.

k) *Find({ other: "YYY" })*, being the attribute indexed.

l) *Find({ other: "YYY" })*, being the attribute NOT indexed.

Note: As typically in RDBMS optimizers, assume uniform distribution of values and statistical independence between pairs of attributes.