I am trying to use mongoDb aggregation pipeline to lookup other table and check if matched some of the attributes.

my query is something like below (p.s. bigtable is sharded):

|  |
| --- |
| db.bigtable.aggregate( [ |
| {$lookup:{ from: "user", localField: |
| "user\_code", foreignField: "code",as: "user"}}, |
| {$match: {"user.country":"US" } }] ) |

I have created index for bigtable.user\_code, user.code & user.country. But the query is very very very slow. I tried explain it, it is COLLSCAN.

It seems to me what using $match with foreign table's attribute, index is not supported and therefore, the query speed is very slow.

**Activity**

* Comments

[Ascending order - Click to sort in descending order](https://jira.mongodb.org/browse/SERVER-28140?actionOrder=desc)

[Permalink](https://jira.mongodb.org/browse/SERVER-28140?focusedCommentId=1516965&page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel#comment-1516965)

[Charlie Swanson](https://jira.mongodb.org/secure/ViewProfile.jspa?name=charlie.swanson) added a comment - Mar 06 2017 06:37:18 PM GMT+0000

Hi all,

Two notes:

1. The query on the 'bigtable' collection is indeed using a collection scan, since there isn't any predicate on that collection. The query on the foreign collection ('user') may well be using an index. There's no great way to tell at the moment, but coming in 3.4.3 you could use $indexStats to determine if the $lookup is using an index (waiting on the fix for ). In general, if the 'user' collection has an index beginning with 'code', I would expect it to choose that index. Currently the $lookup stage invokes the planner on every lookup, so it will behave as if it's executing the query {code: {$eq: <value of user\_code>}}. Unfortunately, the plan chosen may depend on the value used, but usually it doesn't. So you can use an explain on a query of this shape to make sure it's using an index.
2. If you want to ensure that the query on the foreign collection is also incorporating the predicate on 'user.country', you'll need to add an $unwind between the $lookup and the $match. Without an $unwind, the semantics are slightly different, since the document {user: [{country: "US"}, {country: "UK"}]} will match the predicate, but the document {user: {country: "UK"}} will not.

|  |
| --- |
| > db.bigtable.explain().aggregate( [ |
| {$lookup:{ from: "user", localField: "user\_code", foreignField: "code",as: "user"}}, |
| {$match: {"user.country":"US" } } |
| ] ) |
| { |
| "stages" : [ |
| { |
| "$cursor" : { |
| "query" : { |
|  |
| }, |
| "queryPlanner" : { |
| "plannerVersion" : 1, |
| "namespace" : "test.bigtable", |
| "indexFilterSet" : **false**, |
| "parsedQuery" : { |
|  |
| }, |
| "winningPlan" : { |
| "stage" : "COLLSCAN", |
| "direction" : "forward" |
| }, |
| "rejectedPlans" : [ ] |
| } |
| } |
| }, |
| { |
| "$lookup" : { |
| "from" : "user", |
| "as" : "user", |
| "localField" : "user\_code", |
| "foreignField" : "code" |
| } |
| }, |
| { |
| "$match" : { |
| "user.country" : "US" |
| } |
| } |
| ], |
| "ok" : 1 |
| } |
| > db.bigtable.explain().aggregate( [ |
| {$lookup:{ from: "user", localField: "user\_code", foreignField: "code",as: "user"}}, |
| {$unwind: "$user"}, |
| {$match: {"user.country":"US" } } |
| ] ) |
| { |
| "stages" : [ |
| { |
| "$cursor" : { |
| "query" : { |
|  |
| }, |
| "queryPlanner" : { |
| "plannerVersion" : 1, |
| "namespace" : "test.bigtable", |
| "indexFilterSet" : **false**, |
| "parsedQuery" : { |
|  |
| }, |
| "winningPlan" : { |
| "stage" : "COLLSCAN", |
| "direction" : "forward" |
| }, |
| "rejectedPlans" : [ ] |
| } |
| } |
| }, |
| { |
| "$lookup" : { |
| "from" : "user", |
| "as" : "user", |
| "localField" : "user\_code", |
| "foreignField" : "code", |
| "unwinding" : { |
| "preserveNullAndEmptyArrays" : **false** |
| }, |
| "matching" : { |
| "country" : { |
| "$eq" : "US" |
| } |
| } |
| } |
| } |
| ], |
| "ok" : 1 |
| } |

|  |
| --- |
| > db.bigtable.explain().aggregate( [ |
| {$lookup:{ from: "user", localField: "user\_code", foreignField: "code",as: "user"}}, |
| {$match: {"user.country":"US" } } |
| ] ) |
| { |
| "stages" : [ |
| { |
| "$cursor" : { |
| "query" : { |
|  |
| }, |
| "queryPlanner" : { |
| "plannerVersion" : 1, |
| "namespace" : "test.bigtable", |
| "indexFilterSet" : **false**, |
| "parsedQuery" : { |
|  |
| }, |
| "winningPlan" : { |
| "stage" : "COLLSCAN", |
| "direction" : "forward" |
| }, |
| "rejectedPlans" : [ ] |
| } |
| } |
| }, |
| { |
| "$lookup" : { |
| "from" : "user", |
| "as" : "user", |
| "localField" : "user\_code", |
| "foreignField" : "code" |
| } |
| }, |
| { |
| "$match" : { |
| "user.country" : "US" |
| } |
| } |
| ], |
| "ok" : 1 |
| } |
| > db.bigtable.explain().aggregate( [ |
| {$lookup:{ from: "user", localField: "user\_code", foreignField: "code",as: "user"}}, |
| {$unwind: "$user"}, |
| {$match: {"user.country":"US" } } |
| ] ) |
| { |
| "stages" : [ |
| { |
| "$cursor" : { |
| "query" : { |
|  |
| }, |
| "queryPlanner" : { |
| "plannerVersion" : 1, |
| "namespace" : "test.bigtable", |
| "indexFilterSet" : **false**, |
| "parsedQuery" : { |
|  |
| }, |
| "winningPlan" : { |
| "stage" : "COLLSCAN", |
| "direction" : "forward" |
| }, |
| "rejectedPlans" : [ ] |
| } |
| } |
| }, |
| { |
| "$lookup" : { |
| "from" : "user", |
| "as" : "user", |
| "localField" : "user\_code", |
| "foreignField" : "code", |
| "unwinding" : { |
| "preserveNullAndEmptyArrays" : **false** |
| }, |
| "matching" : { |
| "country" : { |
| "$eq" : "US" |
| } |
| } |
| } |
| } |
| ], |
| "ok" : 1 |
| } |

I am trying to use mongoDb aggregation pipeline to lookup other table and check if matched some of the attributes.

my query is something like below (p.s. bigtable is sharded):

db.bigtable.aggregate( [

{$lookup:{ from: "user", localField:

"user\_code", foreignField: "code",as: "user"}},

{$match: {"user.country":"US" } }] )

I have created index for bigtable.user\_code, user.code & user.country. But the query is very very very slow. I tried explain it, it is COLLSCAN.

It seems to me what using $match with foreign table's attribute, index is not supported and therefore, the query speed is very slow.

Activity

Comments

Ascending order - Click to sort in descending order

Permalink

charlie.swanson

Charlie Swanson added a comment - Mar 06 2017 06:37:18 PM GMT+0000

Hi all,

Two notes:

The query on the 'bigtable' collection is indeed using a collection scan, since there isn't any predicate on that collection. The query on the foreign collection ('user') may well be using an index. There's no great way to tell at the moment, but coming in 3.4.3 you could use $indexStats to determine if the $lookup is using an index (waiting on the fix for SERVER-27253). In general, if the 'user' collection has an index beginning with 'code', I would expect it to choose that index. Currently the $lookup stage invokes the planner on every lookup, so it will behave as if it's executing the query {code: {$eq: <value of user\_code>}}. Unfortunately, the plan chosen may depend on the value used, but usually it doesn't. So you can use an explain on a query of this shape to make sure it's using an index.

If you want to ensure that the query on the foreign collection is also incorporating the predicate on 'user.country', you'll need to add an $unwind between the $lookup and the $match. Without an $unwind, the semantics are slightly different, since the document {user: [{country: "US"}, {country: "UK"}]} will match the predicate, but the document {user: {country: "UK"}} will not.

> db.bigtable.explain().aggregate( [

{$lookup:{ from: "user", localField: "user\_code", foreignField: "code",as: "user"}},

{$match: {"user.country":"US" } }

] )

{

"stages" : [

{

"$cursor" : {

"query" : {

},

"queryPlanner" : {

"plannerVersion" : 1,

"namespace" : "test.bigtable",

"indexFilterSet" : false,

"parsedQuery" : {

},

"winningPlan" : {

"stage" : "COLLSCAN",

"direction" : "forward"

},

"rejectedPlans" : [ ]

}

}

},

{

"$lookup" : {

"from" : "user",

"as" : "user",

"localField" : "user\_code",

"foreignField" : "code"

}

},

{

"$match" : {

"user.country" : "US"

}

}

],

"ok" : 1

}

> db.bigtable.explain().aggregate( [

{$lookup:{ from: "user", localField: "user\_code", foreignField: "code",as: "user"}},

{$unwind: "$user"},

{$match: {"user.country":"US" } }

] )

{

"stages" : [

{

"$cursor" : {

"query" : {

},

"queryPlanner" : {

"plannerVersion" : 1,

"namespace" : "test.bigtable",

"indexFilterSet" : false,

"parsedQuery" : {

},

"winningPlan" : {

"stage" : "COLLSCAN",

"direction" : "forward"

},

"rejectedPlans" : [ ]

}

}

},

{

"$lookup" : {

"from" : "user",

"as" : "user",

"localField" : "user\_code",

"foreignField" : "code",

"unwinding" : {

"preserveNullAndEmptyArrays" : false

},

"matching" : {

"country" : {

"$eq" : "US"

}

}

}

}

],