This document shows a query clause which causes Mongo to sequentially scan documents although a much more efficient indexed query is available.

In brief, a large number of nScannedObjects occurs when a sharded $in query contains both a limit() and sort() clause. Bellow is a console walk through that clearly shows the issue:

// Connecting to our v1.8.2-rc0 database

```
davidwas@dev-3:/$ mongo mongotest-shard-0-a:16090/foursquare
MongoDB shell version: 1.8.2-rc0
connecting to: mongotest-shard-0-a:16090/foursquare
```

// We have a collection of checkins which is sharded by uid (user id)

```
> db.checkins.stats()
{
      "sharded" : true,
      "flags" : 1,
      "ns" : "foursquare.checkins",
      "indexSizes" : {
            "_id_" : ...,
            "uid_1" : ...,
            "uid_1__id_-1" : ...,
            "venueid_1__id_-1_uid_1" : ...
      },
      "shards" : {
            "staging0" : {
                  "ok" : 1
            },
            "staging1" : {
                  "ok" : 1
            }
      },
      "ok" : 1
}
```

// Now for the simple query.
// Let us look for a single user's most recent 2 check-ins.
// We have an index on {uid: 1, _id: -1} so this should be rather fast, and it is.

```
> db.checkins.find({uid: {$in: [32]}}).limit(2).sort({_id: -1}).explain()
{
      "clusteredType" : "ParallelSort",
      "shards" : {
            "staging1/mongotest-shard-0-a:17018,mongotest-shard-0-
c:17018,mongotest-shard-0-b:17018" : [
                  {
                        "cursor" : "BtreeCursor uid_1__id_-1",
                        "nscanned" : 2,
                        "nscannedObjects" : 2,
                        "n" : 2,
                        "millis" : 0,
```

```
                        "nYields" : 0,
                        "nChunkSkips" : 0,
                        "isMultiKey" : false,
                        "indexOnly" : false,
                        "indexBounds" : {
                              "uid" : [
                                    [
                                          32,
                                          32
                                    ]
                              ],
                              "_id" : [
                                    [
                                          {
                                                "$maxElement" : 1
                                          },
                                          {
                                                "$minElement" : 1
                                          }
                                    ]
                              ]
                        }
                  }
            ]
      },
      "n" : 2,
      "nChunkSkips" : 0,
      "nYields" : 0,
      "nscanned" : 2,
      "nscannedObjects" : 2,
      "millisTotal" : 0,
      "millisAvg" : 0,
      "numQueries" : 1,
      "numShards" : 1
}
```

// Now let us do the same query as a IN query for two different users.
// Each user is located on a different shard.
// We should have the same performance in which nscanned is rather low

```
> db.checkins.find({uid: {$in: [32,400438]}}).limit(2).sort({_id: -1}
).explain()
{
      "clusteredType" : "ParallelSort",
      "shards" : {
            "staging0/mongotest-shard-0-c:17017,mongotest-shard-0-
b:17017,mongotest-shard-0-a:17017" : [
                  {
                        "cursor" : "BtreeCursor uid_1 multi",
                        "nscanned" : 1103,
                        "nscannedObjects" : 1102,
                        "n" : 2,
                        "scanAndOrder" : true,
                        "millis" : 15,
                        "nYields" : 0,
                        "nChunkSkips" : 0,
```

```
                        "isMultiKey" : false,
                        "indexOnly" : false,
                        "indexBounds" : {
                              "uid" : [
                                    [
                                          32,
                                          32
                                    ],
                                    [
                                          400438,
                                          400438
                                    ]
                              ]
                        }
                  }
            ],
            "staging1/mongotest-shard-0-a:17018,mongotest-shard-0-
c:17018,mongotest-shard-0-b:17018" : [
                  {
                        "cursor" : "BtreeCursor _id_ reverse",
                        "nscanned" : 3920,
                        "nscannedObjects" : 3920,
                        "n" : 2,
                        "millis" : 58,
                        "nYields" : 0,
                        "nChunkSkips" : 0,
                        "isMultiKey" : false,
                        "indexOnly" : false,
                        "indexBounds" : {
                              "_id" : [
                                    [
                                          {
                                                "$maxElement" : 1
                                          },
                                          {
                                                "$minElement" : 1
                                          }
                                    ]
                              ]
                        }
                  }
            ]
      },
      "n" : 4,
      "nChunkSkips" : 0,
      "nYields" : 0,
      "nscanned" : 5023,
      "nscannedObjects" : 5022,
      "millisTotal" : 73,
      "millisAvg" : 36,
      "numQueries" : 2,
      "numShards" : 2
}
>
```

// I attempted this in production, doing a query for the most recent check-in of my 149 friends.
// I expected the {uid: 1, _id: -1} index to reach each of my 149 friend's most recent check-in.
// Thus at most 149 nscannedObjects
// The prod query showed 55,650 nscannedObjects

// I then queried for the the most recent 200 check-ins of my 149 friends.
// I would expect at most 29,800 nScannedObjects(149 friends x 200 most recent)
// The prod query showed 230,174 nscannedObjects
// This is the same number as the total number of all check-ins held by my friends 230,175