## **Advanced Databases**

## Cassandra CA

Student Number: D20125299

Student Name: Luke Hallinan

**Programme Code:** TU856

1. Setting up the cluster and keyspace

We have a cluster with two nodes.

```
|/ State=Normal/Leaving/Joining/Moving
-- Address Load Tokens Owns (effective) Host ID Rack
UN 172.17.0.3 215.92 KiB 16 100.0% b589c151-9fd0-4074-a4be-9943cab8a432 rack1
UN 172.17.0.2 203.95 KiB 16 100.0% 4f96e4e6-ad9f-46b2-bb0c-ddcbe701f16f rack1
```

Both nodes are up and functioning normally. Very little load on either only about 200 Kib and each has a unique host ID should It be needed

We have made a keyspace with a simple strategy that has 2 copies for backup and restore purposes.

2. Porting the data to Cassandra

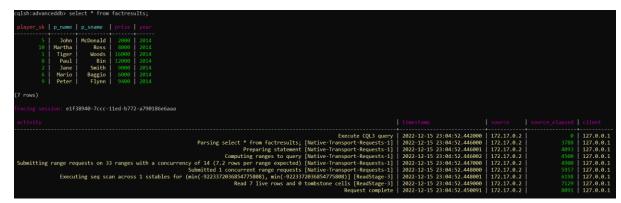
```
|/ State=Normal/Leaving/Joining/Moving
-- Address Load Tokens Owns (effective) Host ID Rack
UN 172.17.0.3 215.92 KiB 16 100.0% b589c151-9fd0-4074-a4be-9943cab8a432 rack1
UN 172.17.0.2 203.95 KiB 16 100.0% 4f96e4e6-ad9f-46b2-bb0c-ddcbe701f16f rack1
```

```
Table: factresults
SSTable count: 1
Old SSTable count: 0
Space used (live): 5412
Space used (total): 5412
Space used by snapshots (total): 0
Off heap memory used (total): 40
SSTable Compression Ratio: 0.6835820895522388
Number of partitions (estimate): 7
Memtable cell count: 0
Memtable data size: 0
Memtable off heap memory used: 0
Memtable switch count: 1
Local read count: 7
Local read latency: NaN ms
Local write count: 28
Local write latency: NaN ms
Pending flushes: 0
Percent repaired: 0.0
Bytes repaired: 0.000KiB
Bytes unrepaired: 0.327KiB
Bytes pending repair: 0.000KiB
Bloom filter false positives: 0
Bloom filter false ratio: 0.00000
Bloom filter space used: 24
Bloom filter off heap memory used: 16
Index summary off heap memory used: 16
Compression metadata off heap memory used: 8
Compacted partition minimum bytes: 43
Compacted partition maximum bytes: 50
Compacted partition mean bytes: 50
Average live cells per slice (last five minutes): NaN
Maximum live cells per slice (last five minutes): 0
Average tombstones per slice (last five minutes): NaN
Maximum tombstones per slice (last five minutes): 0
Dropped Mutations: 0
Droppable tombstone ratio: 0.00000
```

Data shows that there are 7 partitions and the read/write count. Also shoes the space used by the tables.

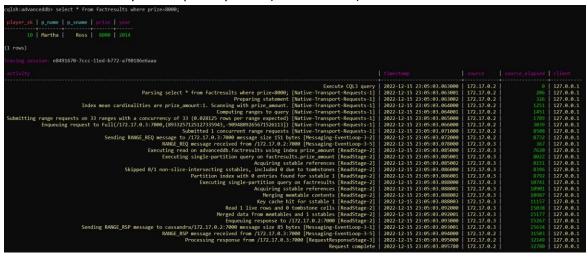
## 3. Golf data

a. Basic query on golf data



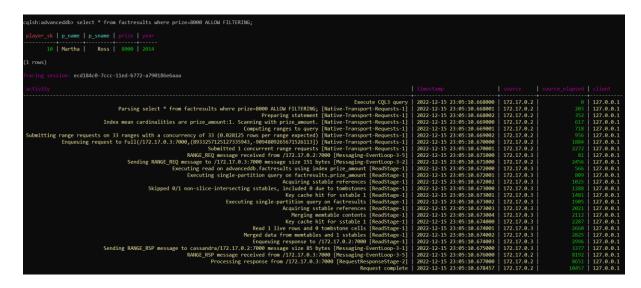
Getting the whole tables took 8091 ms

b. Query of non-primary index (without index)



Much longer to execute, 32780 ms. Checks all 33 ranges rather then just the prize column.

c. Adding a secondary index to golf data

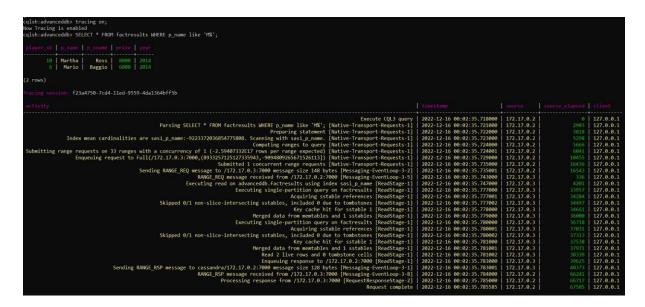


Much faster then before the index was added only 10457 ms.

```
Table (index): factresults.prize_amount
SSTable count: 1
Old SSTable count: 0
Space used (live): 5416
Space used (total): 5416
Space used by snapshots (total): 0
Off heap memory used (total): 40
SSTable Compression Ratio: 0.6737804878048781
Number of partitions (estimate): 7
Memtable cell count: 0
Memtable data size: 0
Memtable off heap memory used: 0
Memtable switch count: 1
Local read count: 0
Local read latency: NaN ms
Local write count: 7
Local write latency: NaN ms
Pending flushes: 0
Percent repaired: 0.0
Bytes repaired: 0.000KiB
Bytes unrepaired: 0.320KiB
Bytes pending repair: 0.000KiB
Bloom filter false positives: 0
Bloom filter false ratio: 0.00000
Bloom filter space used: 24
Bloom filter off heap memory used: 16
Index summary off heap memory used: 16
Compression metadata off heap memory used: 8
Compacted partition minimum bytes: 43
Compacted partition maximum bytes: 50
Compacted partition mean bytes: 50
Average live cells per slice (last five minutes): NaN
Maximum live cells per slice (last five minutes): 0
Average tombstones per slice (last five minutes): NaN
Maximum tombstones per slice (last five minutes): 0
Dropped Mutations: 0
Droppable tombstone ratio: 0.00000
```

Table index now shows in stablestats. Takes up 5416 bytes so more space is needed for each index the database has.

d. Adding an SASI index to golf data to facilitate pattern matching



This has allowed us to fins everyone with a name starting with M which would otherwise have been very difficult. It does however take significant amounts of time compared to other searches at 67585ms.

This method does not create a new table index like the normal manual indexing does. This saves on space but still allows like searches.

## 4. Data including collection data type

```
| Clarks down code | California | Control | Cash |
```

This allows us to search more effectively with only 19148ms for the full result.

```
Table (index): jobhist.jobs_idx
SSTable count: 1
Old SSTable count: 0
Space used (live): 5390
Space used (total): 5390
Space used by snapshots (total): 0
Off heap memory used (total): 35
SSTable Compression Ratio: 0.6194029850746269
Number of partitions (estimate): 4
Memtable cell count: 0
Memtable data size: 0
Memtable off heap memory used: 0
Memtable switch count: 1
Local read count: 0
Local read latency: NaN ms
Local write count: 6
Local write latency: NaN ms
Pending flushes: 0
Percent repaired: 0.0
Bytes repaired: 0.000KiB
Bytes unrepaired: 0.262KiB
Bytes pending repair: 0.000KiB
Bloom filter false positives: 0
Bloom filter false ratio: 0.00000
Bloom filter space used: 16
Bloom filter off heap memory used: 8
Index summary off heap memory used: 19
Compression metadata off heap memory used: 8
Compacted partition minimum bytes: 43
Compacted partition maximum bytes: 86
Compacted partition mean bytes: 71
Average live cells per slice (last five minutes): NaN
Maximum live cells per slice (last five minutes): 0
Average tombstones per slice (last five minutes): NaN
Maximum tombstones per slice (last five minutes): 0
Dropped Mutations: 0
Droppable tombstone ratio: 0.00000
```

This method also add a new index table which takes up more space. For a small table like this the index takes up almost as much space as it though it does improve performance.