Пояснительная записка к домашнему заданию по теме Индексы

# Введение

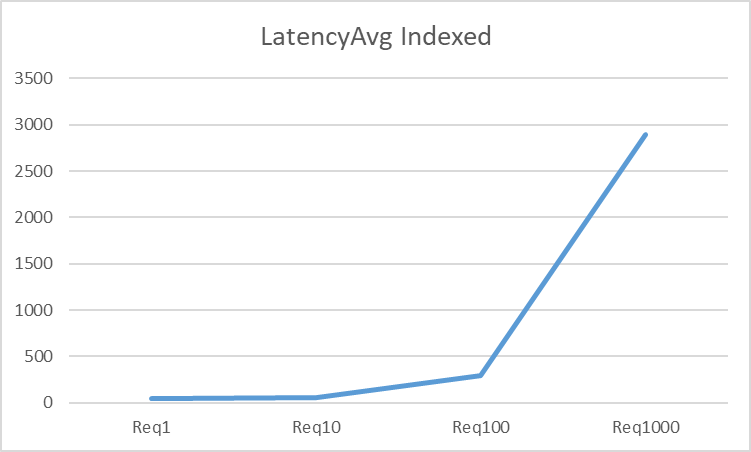
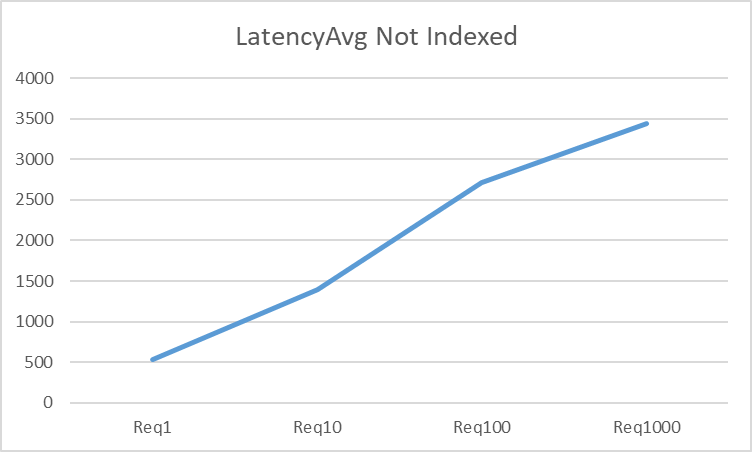
Для реализации поиска анкет по фамилии и имени использован составной индекс по двум полям: *LastName* и *FirstName* (*CREATE INDEX IX\_LASTNAME\_FIRSTNAME ON USERS (LastName, FirstName)*). В качестве первого столбца выбрано поле *LastName*, поскольку обладает большей селективностью (имена повторяются чаще, чем фамилии). Важно, чтобы в запросе в предложении WHERE критерий по столбцам соответствовал порядку в индексе, например, *WHERE LastName LIKE ‘Smith’ AND FirstName LIKE ‘Bob’*, а не наоборот. Можно было бы ограничить количество символов индексируемых полей, используемых при построении индекса – это позволит уменьшить размер индекса без значительной потери производительности. Это целесообразно сделать после заполнения базы данных на регулярной основе, и при необходимости перестроить индекс.

# Результаты тестирования

Ниже представлены результаты тестирования с использованием инструмента WRK

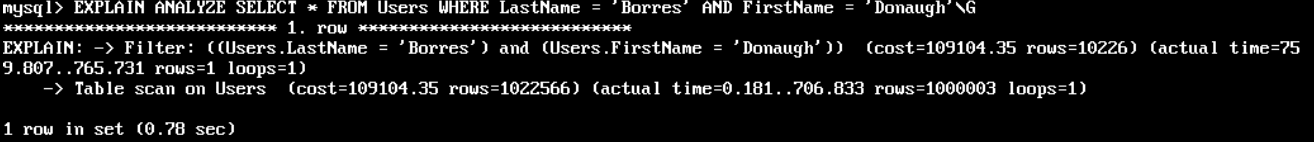
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Avg (ms) | Stdev (ms) | Max (ms) | +/-Stdev (%) | Request/sec | Transfer/sec (KB) |
| **Without Index** |  |  |  |  |  |  |
| Latency |  |  |  |  |  |  |
| Req1 | 530.71 | 4910 | 541.16 | 65.22 | 1.56 | 0.40844 |
| Req10 | 1400 | 234.27 | 2020 | 69.27 | 6.86 | 1.75 |
| Req100 | 2710 | 1240 | 4990 | 50 | 7.02 | 1.79 |
| Req1000 | 3440 | 100 | 4990 | 33.33 | 6.78 | 1.73 |
| Req/Sec |  |  |  |  |  |  |
| Req1 | 1.57 | 0.54 | 2 | 59.57 |  |  |
| Req10 | 9.11 | 7.48 | 70 | 86.01 |  |  |
| Req100 | 7.65 | 6.18 | 40 | 73.05 |  |  |
| Req1000 | 9.89 | 9.26 | 60 | 81.82 |  |  |
| **Indexed** |  |  |  |  |  |  |
| Latency |  |  |  |  |  |  |
| Req1 | 40.71 | 58.32 | 192.82 | 79.76 | 341.09 | 86.94 |
| Req10 | 58.02 | 65.69 | 206.91 | 76.73 | 344.56 | 87.82 |
| Req100 | 290.4 | 85 | 467 | 70.92 | 343 | 87.43 |
| Req1000 | 2890 | 279.45 | 3770 | 92.05 | 323.17 | 82.37 |
| Req/Sec |  |  |  |  |  |  |
| Req1 | 354.18 | 212.32 | 192.82 | 79.76 |  |  |
| Req10 | 34.82 | 14.99 | 80 | 76.91 |  |  |
| Req100 | 34.63 | 15.49 | 101 | 80.71 |  |  |
| Req1000 | 88.8 | 112.6 | 590 | 84.82 |  |  |
|  |  |  |  |  |  |  |

Ниже представлены графики на основе табличных данных



На скриншотах ниже представлены Explain запросов (первый – таблица индексирована, второй - нет).





В случае, если индекс задан, cost=0.35, в противном случае – cost=109104. Разница значительная. Наличие индекса драматически увеличивает производительность.

Следует отметить, что при разработке учебного приложения, я уже использовал индексы. При отображении сообщений по автору и комментариев к сообщению приложение просто висело. Сделал индекс по внешнему ключу MessageID таблицы комментариев и индекс по внешнему ключу AuthorID таблицы сообщений – приложение взлетело.