ВПЛИВ ЗБІРКИ MONGODB З ВИХІДНОГО КОДУ НА ЇЇ ПРОДУКТИВНІСТЬ

НТУ «Дніпровська політехніка»

Максименко Олександр Володимирович
Наукові керівники: **к.т.н., доц. Приходченко С.Д., к.т.н., доц. Ширін А.Л.**

Інформатика як наука займається рядом питань, одне з яких можна сформулювати наступним чином: “Як можна вирішувати ти чи інші обчислювальні та інформаційні задачі максимально ефективно?”. [1] Це питання можна розділити на два: “Як зробити так, щоб програма працювала швидше?” та “Як зробити так, щоб програма споживала менше ресурсів?”. Відомо, що чим швидше працює програма, тим більше оперативної пам’яті вона споживає [2]. Одже, на комп’ютерах з обмеженими ресурсами програмні продукти працюватимуть повільно, однак чи можна якось вплинути на це, не змінюючи алгоритми програми та не збільшуючи ресурсні можливості комп’ютеру? З цією метою було проведено дослідження, в якому було зібрано базу даних MongoDB з вихідного коду (далі — база X), виконано заміри споживаних нею ресурсів процесора та ОЗУ, а також часу, необхідного для виконання типових операцій (вставки, оновлення, видалення, пошуку). Потім було встановлено MongoDB з готового бінарного файлу з офіційного сайту (далі — база Y) та повторено усі заміри. Для наочності було використано набір даних розміром 500000 документів. У якості інструменту моніторингу було використано утиліту htop [3]. Дослідження проводилось на комп’ютері з 8 Гб ОЗУ, 4-х ядерним процесором Intel Core i5 та ОС Archlinux.

У результаті дослідження було виявлено, що вже після старту бази версія із готового бінарного файлу споживає більше оперативної пам’яті та більше ресурсів процесора (рис. 1 та рис. 2).

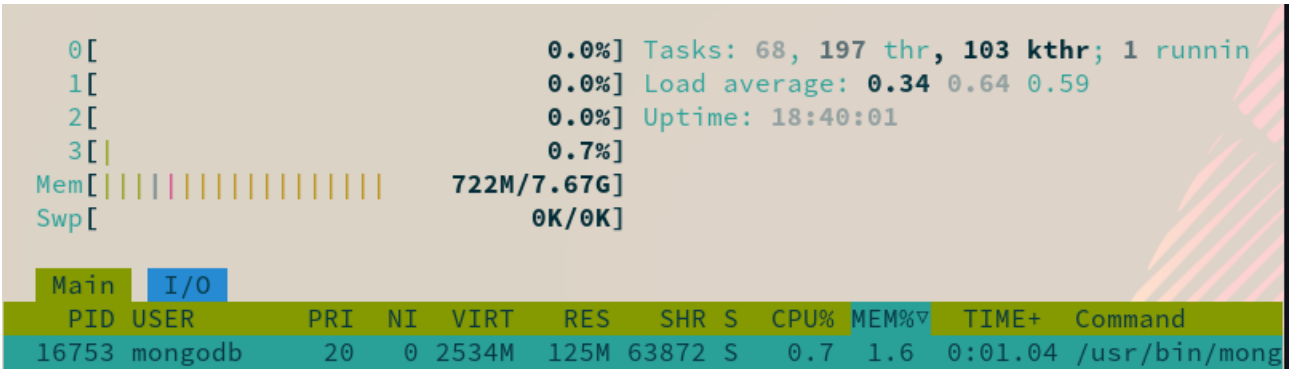

Рис. 1 — Вивід htop після старту бази Y

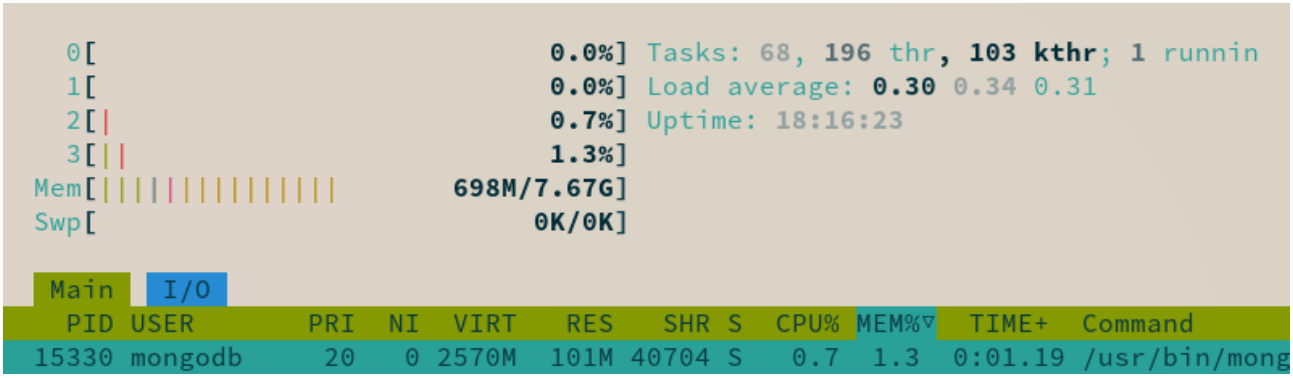

Рис. 2 — Вивід htop після старту бази X

При навантаженні з імпортуванням 500000 документів, база X споживала в середньому на 1.2 % менше оперативної пам’яті, ніж база Y. Також меншою була завантаженість CPU.

Іншим чином виглядає картина при аналізі швидкості виконання операцій. Так, при виконанні операцій запису (insert, update, delete), база Y була швидшою на 13-28% (рис. 3 та рис. 4).

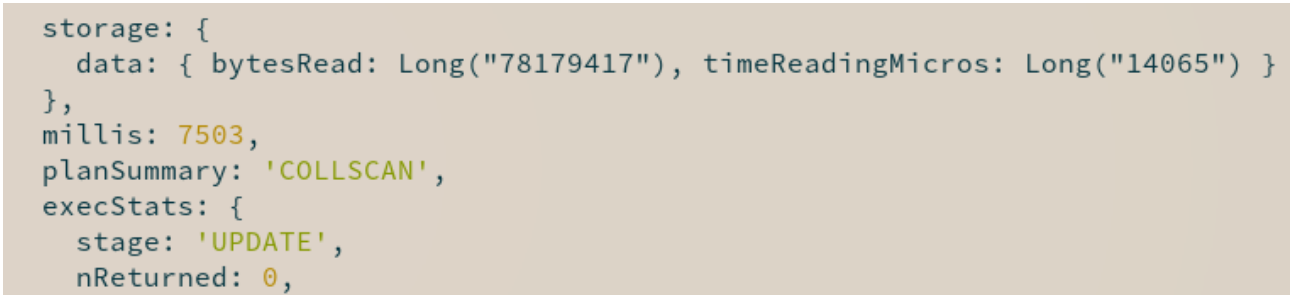

Рис. 3 — Час виконання оновляння у базі Y

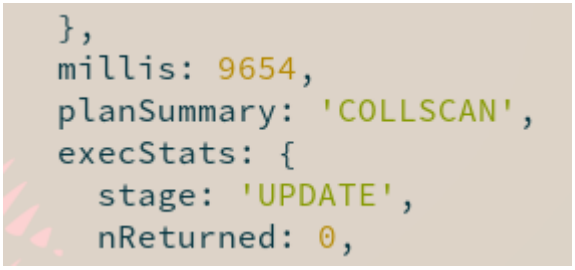

Рис. 4 — Час виконання оновлення у базі X

При виконанні операцій зчитування база X демонструвала або однакові результати з базою Y, або була трохи швидшою (до 9%) (рис. 5 та рис. 6).

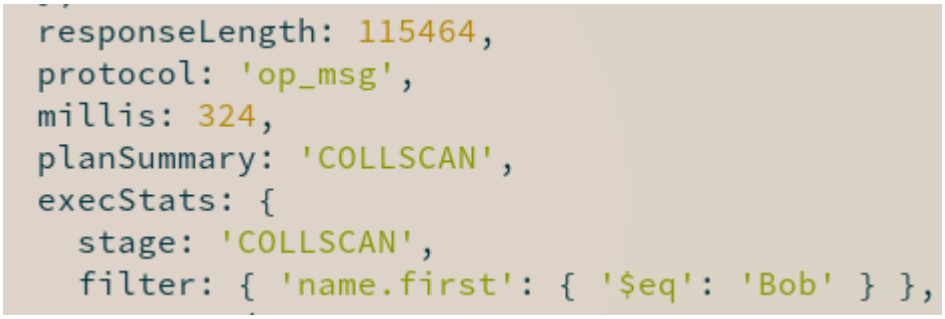

Рис. 5 — Час виконання операції читання у базі Y

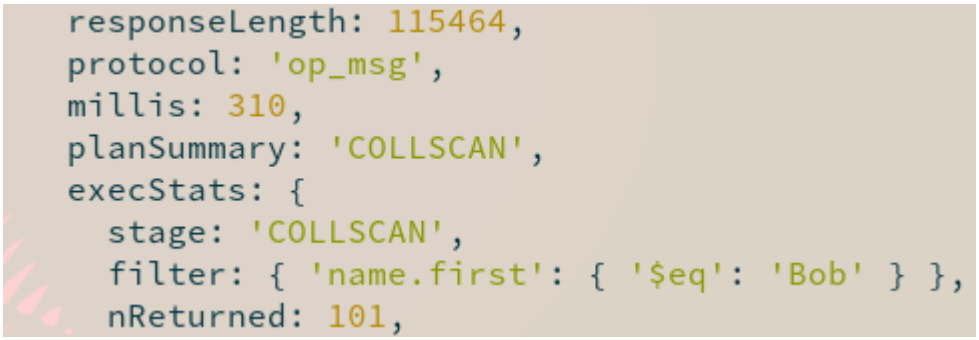

Рис. 6 — Час виконання операції читання у базі X

У якості висновку можна стверджувати, що при використанні MongoDB на комп'ютерах з обмеженими ресурсами є сенс збирати її з вихідного коду для зменшення навантаження на ОЗУ та процесор. Також дослідження має наступні перспективи:

- проведення на комп'ютерах з різним "залізом" та різними ОС;
- вивчення можливостей використання різних параметрів компілятору для оптимізації процесу збірки та вихідних виконуваних файлів під конкретну платформу.

Перелік посилань

1. Computer science: Вікіпедія [Електронний ресурс]. URL: https://en.wikipedia.org/wiki/Computer_science *(дата звернення: 15.04.2023).*

2. How Random Access Memory (RAM) affets performance: Dell Technologies [Електронний ресурс]. URL: https://www.dell.com/support/kbdoc/ru-ua/000129805/how-random-access-memory-ram-affects-performance *(дата звернення: 15.04.2023).*

3. Pēteris Ņikiforovs, htop explained [Електронний ресурс]. URL: https://peteris.rocks/blog/htop/ *(дата звернення: 15.04.2023).*