**Người soạn: Lê Huy**

**Ngày soạn: 04/01/2024**

**Topic: Export 10 triệu record ra CSV**

**BÁO CÁO**

1. **Yêu cầu:**

Export toàn bộ dữ liệu trên ra định dạng CSV, nhanh nhất có thể.

Viết ra 1 phương án thơ ngây nhất, tìm hiểu có vấn đề gì ko, sau đó cải thiện dần, đưa ra thời gian chạy của từng phương án, sau mỗi lần cải thiện thì tăng lên được bao nhiêu?

1. **Tổng hợp kết quả (xếp theo thời gian thực hiện từ chậm đến nhanh):**

*Ghi chú: Kết quả thời gian là lần test có kết quả nhanh nhất đối với một phương án.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Phương án** | **Thời gian**  **(giây)** | **Toàn vẹn dữ liệu** | **Bảo mật** | **Ưu điểm** | **Nhược điểm** | **Đề xuất** |
| Batched Export Prepared Statement | 295.001 | Ok | Ok | An toàn đối với SQL Injection, câu query được compile 1 lần | Tốc độ vẫn không tối ưu bằng batched export | Sử dụng khi có yêu cầu security |
| Batched Export Method | 272.021 | Bình thường | Bình thường | Thời gian nhanh, đơn giản | Nguy cơ SQL injection | Sử dụng khi yêu cầu bảo mật được ưu tiên thấp |
| Select Into Outfile | 66.480 | Thấp | Bình thường  (raw data) | Nhanh nhất | Bảo mật bình thường, toàn vẹn dữ liệu thấp | Sử dụng khi dữ liệu lớn và đã được chuẩn hóa, ưu tiên tốc độ |

1. **Chi tiết:**
2. **Nhận định một số yếu tố ảnh hưởng đến hiệu suất:**

Khi export số lượng lớn record từ MySQL ra file CSV sử dụng PHP; tương tự quá trình import, có nhiều yếu tố có thể ảnh hưởng đến hiệu suất như:

* Cấu trúc dữ liệu, ràng buộc.
* Phần cứng: CPU, Memory, RAM, tốc độ I/O của ổ cứng.
* Tài nguyên server.
* Tình trạng network.
* Thông số server, MySQL, PHP.
* *Batch-size của từng câu lệnh.*
* *Ảnh hưởng của transaction.*
* *Các phương thức tối ưu câu query.*

Trong phạm vi tìm hiểu và khả năng, trong báo cáo sẽ thử nghiệm một số phương án xoay quanh các các yếu tố batch-size của từng câu lệnh, ảnh hưởng của transaction và các phương thức tối ưu câu query.

1. **Quá trình thử nghiệm và tối ưu:**

* **Tóm tắt:**
* Triển khai import 10 triệu record từ file user.csv vào database.
* Lựa chọn sử dụng MySQLi để thao tác với MySQL Server để tận dụng những tính năng được tối ưu cho riêng MySQL, bao gồm hiệu suất và tốc độ.
* Đọc dữ liệu từ database và export ra file exported\_user\_data.csv.
* Với mỗi phương án, sử dụng hàm microtime() để tính toán thời gian chạy từ bắt đầu cho đến kết thúc quá trình.
* Test từ phương án đơn giản nhất và xoay quanh các yếu tố tác động. Từ đó tìm các cách để tối ưu.
* **Quá trình thực hiện như sau:**
  1. **Test data volume tải của server:**
* **Phương pháp:**
  + Kiểm tra memory\_limit default là 300MB.
  + Test từng lần export với số lượng record khác nhau, kết hợp với check memory\_peak\_usage để xác định mức record tối đa mà hệ thống có thể tải.
  + Tại mỗi script, fetch data từ MySQL và sử dụng vòng lặp while để fputcsv() xuất ra file csv.
* **Kết quả:** Hệ thống có thể export tối đa khoảng 765,000 record/lần/file.
* **Hiệu suất:** 
  + Với các thông số đã được thiết lập sẵn, hệ thống chỉ có thể tải tối đa khoảng 765,000 record/lần/file. Các mức lớn hơn đều gây tràn memory và khiến MySQL server bị crash.
  + Thông số trên để làm cơ sở test quá trình export 10 triệu record bằng cách chia nhỏ ra thành các mức batch-size khác nhau để tối ưu.
* **File:** export\_data\_test.php
  1. **Export theo từng batch (Batched Export Method):**
* **Phương pháp:**
  + Sử dụng SELECT query kết hợp LIMIT và OFFSET để có thể tùy chỉnh batch cho mỗi lần query.
  + Sử dụng vòng lặp while để fetch data theo từng batch và fputcsv() xuất ra file csv.
  + Test với thông số batch khác nhau (100,000 ~ 765,000 record) để tối ưu.
* **Kết quả:**

|  |  |
| --- | --- |
| **Batch-size** | **Kết quả** |
| 100,000 records | 1150.259 giây |
| 200,000 records | 606.317 giây |
| 300,000 records | 424.322 giây |
| 400,000 records | 320.012 giây |
| 500,000 records | 306.756 giây |
| 550,000 records | 304.117 giây |
| 580,000 records | 279.051 giây |
| **590,000 records** | **272.021 giây** |
| 595,000 records | 274.800 giây |
| >600.000 records | Out of memory |

* **Hiệu suất:** 
  + Hiệu suất về tốc độ export tăng tương ứng với việc tăng batch-size tiệm cận mức tối đa. Kết quả tốc độ nhanh nhất ở batch-size 590,000 records. Hiệu suất được tối ưu với việc tận dụng tối đa memory để tăng batch-size, qua đó giảm thiểu overhead cost của quá trình kết nối, fetch data và ghi file.
  + Càng gần về mức tối đa thì tỷ lệ tối ưu được càng thấp.
  + Tuy nhiên, phương pháp này còn một số điểm có thể tối ưu bao gồm câu query còn sử dụng nhiều lần, nên thử test prepared statement?
* **File:** export\_data.php
  1. **Export theo từng batch kết hợp prepared statement (Batched Export using Prepared Statement):**
* **Phương pháp:**
  + Sử dụng Batched Export Method nêu trên với các thông số batch-size tương tự.
  + Sử dụng prepared statement đối với câu query SELECT.
  + Sử dụng vòng lặp while để fetch data theo từng batch và fputcsv() xuất ra file csv.
* **Kết quả:**

|  |  |
| --- | --- |
| **Batch-size w Prepared Stmt** | **Kết quả** |
| 500,000 records | 302.231 giây |
| 550,000 records | 297.853 giây |
| **555,000 records** | **295.001 giây** |
| 560,000 records | Out of memory |
| 580,000 records | Out of memory |
| 590,000 records | Out of memory |

* **Hiệu suất:** 
  + Hiệu suất về tốc độ có cải thiện và đạt tối ưu ở mức batch-size 555,000 record. Mức tối ưu này cũng không bằng phương án không sử dụng prepared statement.
  + Bắt đầu từ batch-size 560,000 record, bắt đầu xảy ra lỗi Out of memory.
  + Mặc dù, prepared statement giúp giảm số lần query, tuy nhiên việc preparing và binding lại làm tăng performance. Trong kịch bản này, số lượng câu query ít nên hiệu suất của prepared statement mang lại không thể bù đắp cho performance tiêu tốn.
  + Tuy nhiên, prepared statement lại giúp tăng mức độ bảo mật, tránh SQL injection.
* **File:** prepared\_statement\_export.php
  1. **SELECT INTO OUTFILE:**
* **Phương pháp:** 
  + Đây là một phương thức đặc biệt mà MySQL hỗ trợ để thao tác load và export data lớn, nhiều record từ MySQL ra file cụ thể.
  + Tuy nhiên, lưu ý thiết lập bảng và câu lệnh phải tương đồng.
* **Kết quả:** 66.480 giây (khoảng 1 phút).
* **Hiệu suất:** 
  + Hiệu suất về tốc độ tối ưu nhất do đã rút ngắn tối thiểu quá trình tương tác giữa server và database.
  + Tuy nhiên, phương án này chỉ phát huy hiệu quả khi dữ liệu đã được chuẩn hóa, thiết lập theo logic để có thể thực hiện phương án này. Lưu ý đến vấn đề security và xung đột kiểu dữ liệu khi áp dụng phương thức này.
* **File:** select\_into\_outfile.php