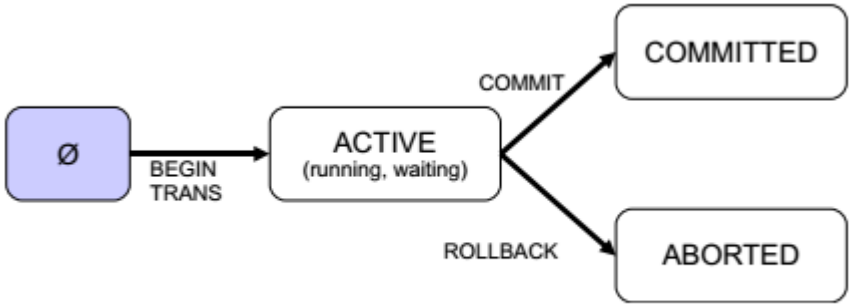
Quản trị cơ sở dữ liệu và Tối ưu hiệu năng

Tối ưu phục hồi
(Recovery Tuning)

Danh mục

- Tối ưu phục hồi (Recovery Tuning)
  - Ghi nhật kí và phục hồi (Logging and Recovery)
  - Tối ưu hệ thống con phục hồi (Tunning the Recovery subsistem)

Tính nguyên tử và tính bền trong trường hợp thất bại



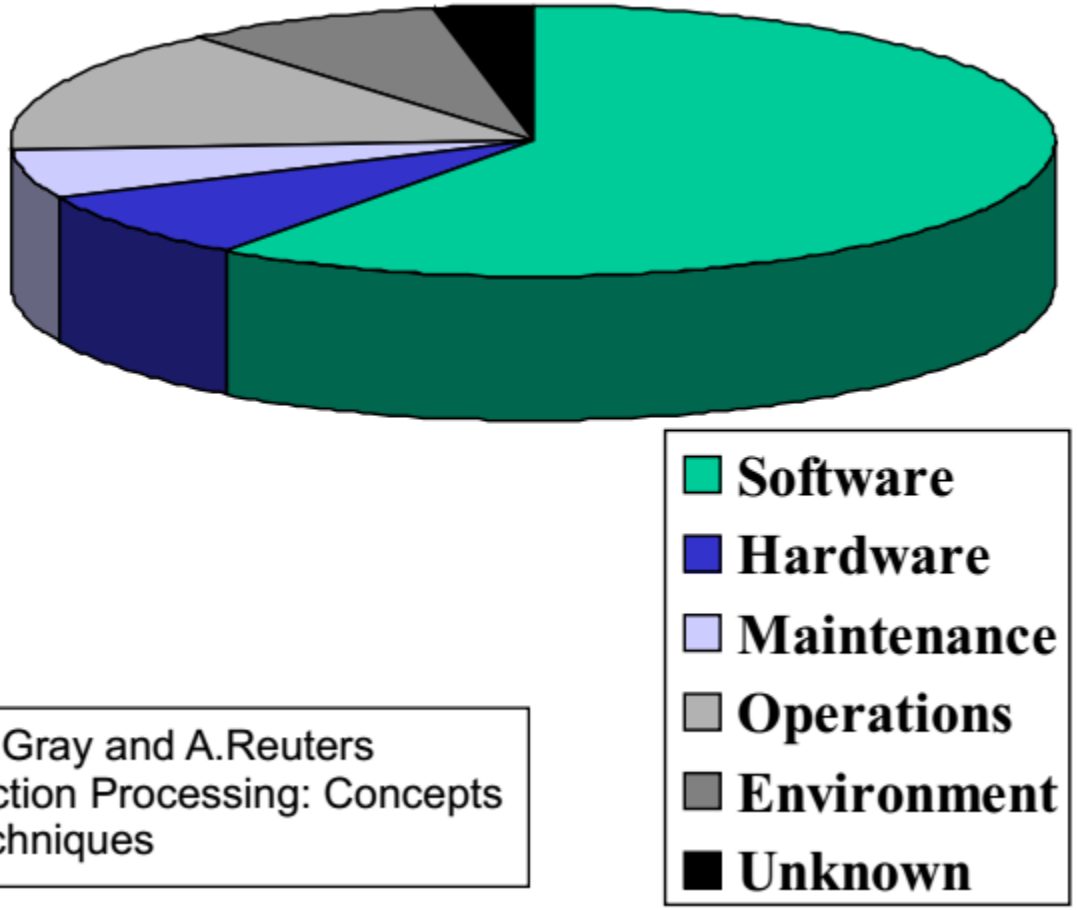
States of a Transaction

- Tính bền (Durability): Sau khi giao dịch cam kết, các thay đổi của CSDL vẫn còn ngay cả khi trường hợp sự cố hệ thống
- Tính nguyên tử (Atomicity): Sau khi thất bại, xây dựng lại CSDL sao cho:
  - Các thay đổi của tất cả các giao dịch đã cam kết được phản ánh lại
  - Các tác động của tất cả các giao dịch chưa cam kết và hủy bỏ bị loại bỏ
- Hệ thống con phục hồi (Recovery subsystem): Đảm bảo tính nguyên tử và tính bền trong trường hợp thất bại

Các loại sự cố (Failure Types)

- Phần mềm (Software):
  - 99% là các Heisenbug (không tái sử dụng lại được(non-reproducible), do sự điều chỉnh (timing) hoặc quá tải (overload))
  - Các heisenbug không xuất hiện nếu hệ thống bị khởi động lại
  - Ví dụ: lỗi do mức độ tách biệt được chọn quá thấp
- Phần cứng (Hardware): Sự cố trong thiết bị vật lý
  - CPU, RAM, ổ cứng, network
  - Dứng thất bại (fail-stop): Thiết bị dừng khi sự cố xuất hiện, ví dụ CPU
- Bảo trì(Maintenance): vấn đề khi hệ thống sửa chữa hoặc bảo trì
  - Ví dụ: khôi phục lại từ sự cố, sau lưu
- Các thao tác (Operations): Các thao tác thường xuyên
  - Quản trị thệ thống và cấu hình thường xuyên
  - Các thao tác người dùng
- Môi trường (Environment): Các yếu tố bên ngoài hệ thống máy tính
  - Ví dụ: Cháy trong phòng máy (Credit Lyonnais, 1996), 9/11

Xác suất sự cố



From J.Gray and A.Reuters
Transaction Processing: Concepts
and Techniques

Các hệ CSDL có thể chịu đựng được những sự cố nào?

- Một vài sự cố phần mềm:
  - Bị treo bên phía client
  - Bị treo hệ điều hành
  - Một vài lỗi phía server
- Thất bại phần cứng:
  - CPU dừng thất bại (fail-stop) và xóa bộ nhớ chính
  - Ổ đĩa đơn dừng thất bại (nếu có đủ ổ đĩa dự phòng)
- Môi trường: Bị cúp điệm
- Hoạt động sau lưu vẫn quan trọng:
  - Hệ thống phục hồi (recovery system) không thay thế được các hoạt động sao lưu
  - Hoạt động sao lưu cần thiết cho các sự cố không được bảo vệ bởi hệ thống phục hồi
  - Ví dụ: Bị xóa bất ngờ, thiên tai

Tính bền (Durability)

- Tính bền trong CSDL:
  - Mục tiêu: Tạo những thay đổi cố định trước khi gửi cam kết đến client
  - Thực hiện: Lưu trữ dữ liệu giao dịch trên kho bền vững (stable storage)
- Kho bền vững: Tránh được sự cố (chỉ gần đúng trong thực hành)
  - Các phương tiện bền vững, ví dụ: ổ cứng, dải (tape), battery-backed RAM
  - Tạo bản sao trên một số đơn vị (Các ổ đĩa dự phòng để chống các sựu cố ổ đĩa)
- Vấn đề:
  - Các bộ đệm không bền (non-durable buffers) trong một vài lớp hệ thống
  - Ghi đĩa không hoàn chỉnh (partial disk writes)

Làm thế nào đối phó với các bộ đệm không bền (Non-durable buffer)

- Bộ đệm không bền (Non-durable buffer) trong một số lớp hệ thống:
  - CSDL cho hệ thống viết một trang ổ đĩa (disk page)
  - Nhưng trang ổ đĩa này vẫn còn trong một số bộ đệm không bền
- Bộ đệm hệ điều hành (Operating system buffer):
  - Các thao tác ghi được đệm
  - fsync làm sạch tất cả các trang của các file đã cho - OK
- Bộ đệm điều khiển ổ đĩa (Disk controller cache):
  - Phổ biến trong các bộ điều khiển RAID
  - battery-backed cache – OK
  - Các cache khác có thể dẫn đến mâu thuẫn trong trường hợp sự cố hệ thống
- Bộ đệm đĩa (Disk cache): Tắt ổ đĩa log (log disk) (Quan trọng!)
  - hdparm -I /dev/sda shows meta data of disk /dev/sda
  - hdparm -W 0 /dev/sda switches disk buffer off

Làm thế nào đối phó với tình trạng ghi đĩa không hoàn chỉnh (Partial disk writes)

- Ghi đĩa không hoàn chỉnh (Partial disk writes):
  - CSDL viết một trang ổ đĩa bao gồm nhiều sector
    Ví dụ: 8kB trang bao gồm 16 sector (mỗi sector: 512B)
  - Sự cố mất điện trong khi viết: Trang có thể bị ghi không hoàn chỉnh
  - Dẫn đến trạng thái CSDL không nhất quán
- Bộ điều khiển ổ đĩa (Disk controller): pin hỗ trợ bộ nhớ cache (battery backed cache)
  - Dữ liệu trong cache được ghi khi khởi động lại sau khi mất điện
  - Trạng thái nhất quán được phục hồi
- Hệ điều hành: hệ thống file
  - Hệ thống file ngăn ngừa việc ghi không hoàn chỉnh, ví dụ: Raiser 4
- Cơ sở dữ liệu: ví dụ: full_page_writes trong PostgreSQL
  - Tiền ảnh (before-image) của trang được lưu trữ trước khi cập nhật nó
  - Phục hồi: Trang ghi không hoàn chỉnh được phục hồi và việc cập nhật được lặp lại

Đảm bảo tính nguyên tử

- Tiền ảnh (Before images): Trạng thái khi giao dịch bắt đầu
  - Được sử dụng để hoàn tác các ảnh hưởng của giao dịch không cam kết
  - Tiền ảnh
- Hậu ảnh (After images): Trạng thái khi giao dịch kết thúc

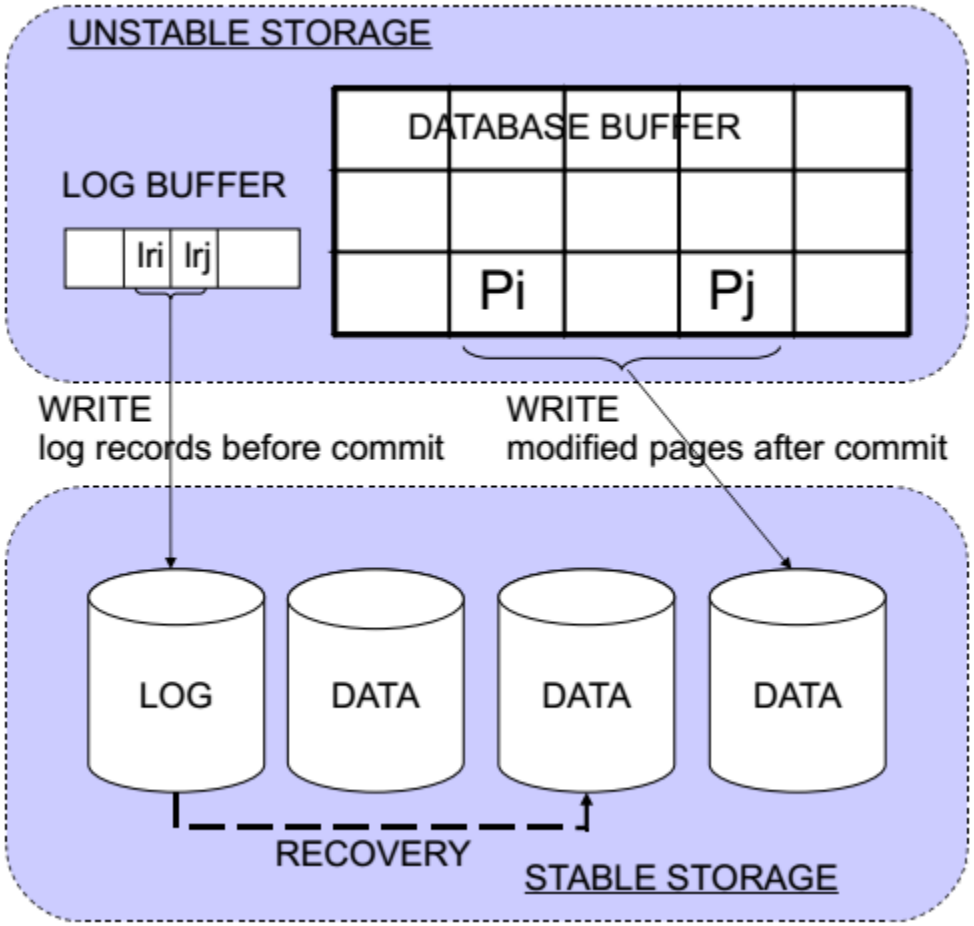
Khái niệm

- File dữ liệu: Các bảng, các chỉ số (indexs)
- File log: Lưu tiền ảnh và hậu ảnh
- Bộ đệm CSDL: Chứa các trang mà các giao dịch sửa đổi
- dirty page: Trang đệm với các thay đổi không được cam kết

Write-Ahead Logging (WAL)

- WAL cam kết (WAL commit):
  - Viết các hậu ảnh đến file log sau khi giao dịch thực hiện cam kết
  - Các file dữ liệu có thể được cập nhật sau (sau khi cam kết)
- WAL hủy bỏ (WAL abort):
  - Biến thể 1: Lưu trữ hậu ảnh trong log
  - Biến thể 2: Sử dụng file dữ liệu như một hậu ảnh
  - Chỉ trong biến thể 1 là an toàn để viết các dirty page đến file dữ liệu.
  - Các dirty page được viết khi bộ đệm CSDL (database buffer) đầy
- Ví dụ: WAL cho giao dịch T thay đổi trang P_i và P_j
  - Trang P_i và P_j được nạp vào bộ đệm CSDL
  - Giao dịch T thay đổi các trang P_i và P_j
  - CSDL sinh ra các bản ghi log Ir_i và Ir_j cho các thay đổi
  - CSDL ghi các bản ghi vào kho bền vững (stable storage) sau khi cam kết
  - Các trang đã thay đổi được ghi vào file dữ liệu sau khi giao dịch T thực hiện cam kết

Write-Ahead Logging (WAL)



Các biến thể logging (Logging Variants)

- Logging granularity: Một bản ghi log lưu trữ cái gì?
  - Logging mức trang (page-level logging)
  - Logging mức byte (byte-level logging hoặc log partial pages)
  - Logging mức bản ghi (record-level logging)
- Logging hợp logic (Logical logging): Thao tác log và argument sinh ra cập nhật
  - Ví dụ: Thao tác: chèn vào employee, argument: (103-4403-33,Brown)
  - Lưu trữ không gian đĩa
  - Implemented trong DB2

Đảm bảo logging (Logging Guarantee)

- Đảm bảo bởi các thuật toán:

  Trạng thái CSDL hiện tại = Trạng thái hiện tại của các file dữ liệu + log
- Trạng thái CSDL hiện tại:
  - Phản ánh tất cả giao dịch đã cam kết
- Trạng thái hiện tại của file dữ liệu:
  - Chỉ phản ánh các giao dịch đã cam kết trong file dữ liệu
  - Một vài giao dịch có thể được cam kết và lưu trữ trong log,nhưng chưa được viết trong CSDL

(Checkpoint and Dump)

- Checkpoint: Bắt buộc các file dữ liệu phải phản ánh trạng thái CSDL hiện tại
  - Ghi tất cả các thay đổi đã cam kết đến file dữ liệu
  - Các thay đổi đã cam kết có thể ở trong bộ đệm CSDL hoặc log
- Khi nào các checkpoint xảy ra?
  - Tại các khoảng chính quy (regular intervals) (tuning parameter)
  - Log bị đầy (Oracle)
  - Lệnh SQL rõ ràng (explicit SQL command)
- Dump: Trạng thái CSDL giao dịch nhất quán (transaction-consistent database state)
  - Toàn bộ CSDL bao gồm những thay đổi của tất cả các giao dịch đã xác nhận
  - Đảm bảo phục hồi:

    Trạng thái CSDL hiện tại = CSDL kết xuất (database dump) + log (sau dump)

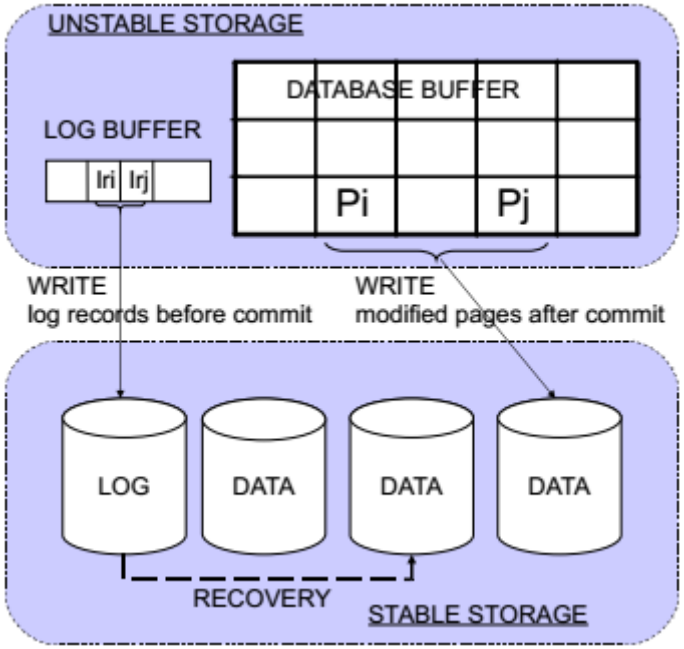
Phục hồi sau khi bộ nhớ chính và ổ cứng gặp sự cố

- Sự cố bộ nhớ chính (Main memory failure): Bộ đệm CSDL bị mất
  - Log cần được xem xét chỉ bắt đầu sau checkpoint cuối cùng
  - Tất cả các thay đổi đã cam kết trước checkpoint đã có trong file dữ liệu
- Sự cố ổ đĩa dữ liệu (Data disk failure): (ổ cứng với log vẫn ổn)
  - Yêu cầu CSDL kết xuất
  - Log sau CSDL kết xuất cần được xem xét
  - Các checkpoint không liên quan
- Sự cố ổ đĩa log (Log disk failure): Thảm họa!
  - Các giao dịch đã cam kết sau checkpoint cuối cùng bị mất
  - CSDL có thể không nhất quán – trạng thái nhất quán cuối cùng là dump cuối cùng
  - Để ngăn ngừa sự cố này, Tạo bản sao ổ đĩa với log
  - Đảm bảo tránh được những rủi ro của cá bộ đệm không bền và việc ghi không hoàn chỉnh.

Danh mục

- Tối ưu phục hồi (Recovery Tuning)
  - Ghi nhật kí và phục hồi (Logging and Recovery)
  - Tối ưu hệ thống con phục hồi (Tunning the Recovery subsistem)

Các hoạt động tối ưu (Tuning Activities)

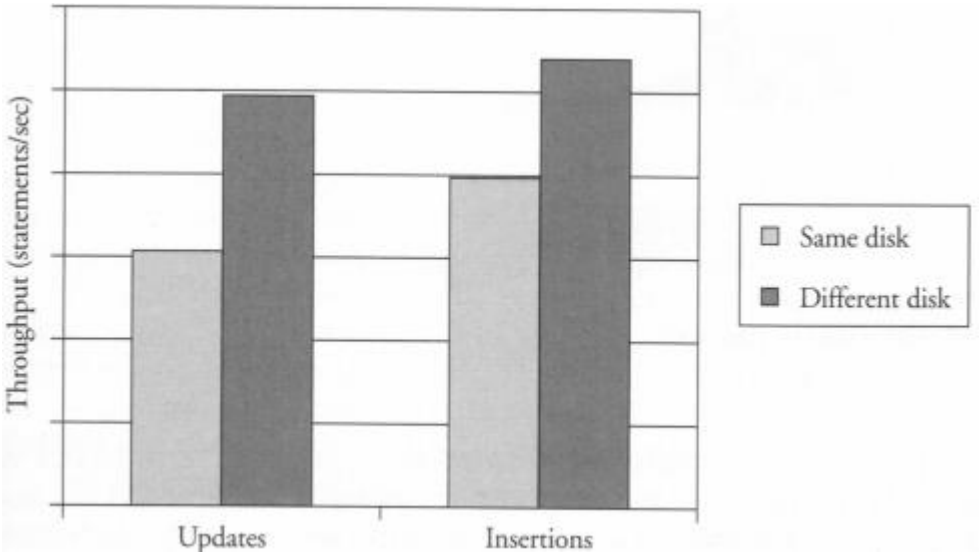


- Log trên ổ đĩa riêng rẽ
- Tối ưu bộ đệm log (Log buffer tuning): Cam kết nhóm (group commit)
- Tối ưu bộ đệm log: Đánh đổi tính bền (trading in durability)
- Tối ưu ghi giữ liệu (checkpoints)

Log trên ổ đĩa riêng rẽ (Log on separate disk)

- Cập nhật giao dịch phải được viết vào log, ví dụ: đến ổ đĩa
- Nếu log và các file dữ liệu chia sẻ ổ đĩa, disk seeks được yêu cầu.
- Ổ đĩa riêng rẽ cho log
  - Ghi tuần tự thay vì seeks (nhanh hơn gấp 10-100 lần)
  - Log độc lập với các file dữ liệu trong trường hợp sự cố ổ cứng
  - Thiết lập ổ đĩa có thể được điều chỉnh đến log (ví dụ: tắt bộ đệm)
- PostgreSQL: Làm thế nào dịch chuyển được log đến các ổ đĩa khác?
  - Đường dẫn log: pg_xlog (location: show data_directory;)
  - Dịch chuyển đường dẫn log đến ổ đĩa log (log disk) và tạo kí tự link (symbolic link)

Thực nghiệm – Log trên ổ đĩa riêng rẽ

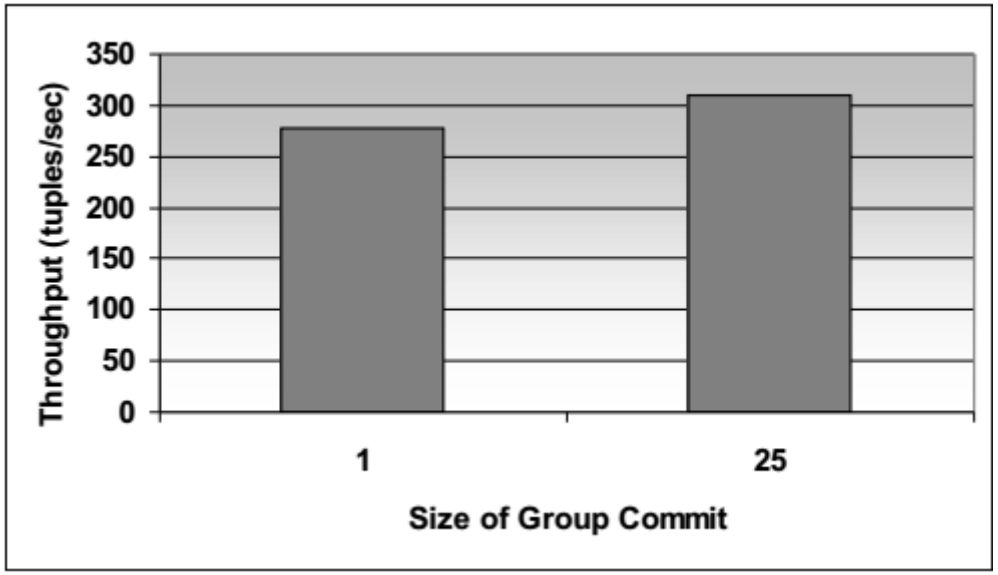


- 300k chèn và cập nhật các báo cáo
- Mỗi báo cáo là một giao dịch riêng biệt và bắt buộc phải có một hoạt động ghi
- Cùng ổ đĩa: Các file dữ liệu và log nằm cùng một ổ cứng.
- Các ổ đĩa khác nhanh: log có ổ cứng riêng

c
(Group Commit)

- Bộ đệm log được làm sạch (flush) ổ đĩa trước mỗi cam kết
- Cam kết nhóm:
  - Cam kết một nhóm của các giao dịch với nhau
  - Chỉ 1 đĩa ghi (làm sạch) cho tất cả giao dịch
- Ưu điểm: Thông lượng cao hơn
- Nhược điểm: Một vài giao dịch phải đợi trước khi cam kết
  - Các khóa bị giữ lâu hơn (cho đến khi cam kết)
  - Thời gian đáp ứng thấp hơn vì phải đợi các giao dịch

Cam kết nhóm – Thực nghiệm



- Khi tăng kích thước cam kết nhóm thì thông lượng cũng tăng

Bộ đệm WAL và cam kết nhóm trong PostgreSQL

- Bộ đệm WAL (WAL buffer):Write ahead log buffer
  - Bộ đệm RAM, kích thước 64kB=8pages (wal_buffers)
  - Tất cả bản ghi log được viết đến bộ đệm này
  - Trang WAL được làm sạch khi cam kết hoặc cứ mỗi 200ms một lần (wal_writer_delay)
  - Dữ liệu được viết đến một file được gọi là khúc WAL (WAL segment)
- Commit_delay: (mặc định: 0)
  - Thời gian trễ giữa một cam kết và làm sạch (flushing) bộ đệm WAL
  - Trong thời gian đợi, hi vọng các giao dịch khác thực hiện cam kết
  - Nếu những giao dịch khác thực hiện cam kết, thực hiện cam kết nhóm
  - Nếu không có các giao dịch khác cam kết, thời gian đợi sẽ mất
- Commit_sibling: (mặc định: 5)
  - Số lượng tối thiểu của các giao dịch mở đồng thời cho nhóm cam kết
  - Nếu các giao dịch được mở ít hơn, commit_delay bị mất tác dụng

3. Tối ưu WAL: Đánh đổi tính bền (PostgreSQL)

- Synchronous_commit: (default: on)
  - Gọi fsync bắt hệ điều hành làm sạch bộ đệm đĩa
  - Chỉ cam kết sau khi fsync returns
  - Tắt nếu bạn không muốn đợi fsync
  - Tham số có thể set cho mỗi giao dịch một cách riêng lẻ
- Tắt đồng bộ cam kết để tăng hiệu năng
- Trường hợp xấu nhất: Tính nhất quán CSDL không còn nguy hiểm
  - Hệ thống bi treo có thể gây ra mất hầu hết các giao dịch đã cam kết gần đây
  - Các giao dịch bị mất có vẻ như không thực hiện cam kết đến CSDL và bị hủy bỏ hết khi khởi động, dẫn đến trạng thái CSDL nhất quán
  - Client nghĩ các giao dịch đã cam kết, nhưng nó đã bị hủy bỏ
  - Trì hoãn tối đa giữa cam kết và làm sạch (giai đoạn rủi ro):
    3 × wal_writer_delay (= 3 × 200ms by default)
- fsync: (default: on)
  - Tắt fsync có thể dẫn đến sai lệch dữ liệu không thể khôi phục
  - synchronous_commit: Hiệu suất tương tự, ít rủi ro

4. Tối ưu ghi dữ liệu (Tuning Data Writes)

- Tại thời gian cam kết
  - Bộ đệm CSDL (trong RAM) đã cam kết thông tin (committed information)
  - Log (trên ổ đĩa) đã cam kết thông tin
  - File dữ liệu có thể không cam kết thông tin
- Tại sao dữ liệu không ghi ngay đến file dữ liệu?
  - Mỗi trang ghi yêu cầu một seek
  - Do ramdom I/O làm hiệu suất bị ảnh hưởng
- Ghi thích hợp (Convenient writes):
  - Đợi và ghi các khối lớn hơn cùng một lúc
  - Ghi khi rẻ , ví dụ: các đầu đĩa (disk heads) ở trên right cylinder

Ghi CSDL – Các tùy chọn tối ưu (Database Writes – Tuning Options)

- Tỉ lệ đầy của bộ đệm CSDL (RAM):
  - Oracle: DB_BLOCK_MAX_DIRTY_TARGET chỉ số lượng tối đa của dirty page trong bộ đệm CSDL
  - SQL Server: Các trang trong danh sách rỗng giảm xuống dưới ngưỡng (mặc định 3%)
- Tần số Checkpoint:
  - Checkpoint bắt buộc tất cả các hoạt động ghi đã cam kết (committed writes) chỉ ở trong bộ đệm CSDL hoặc log đến file dữ liệu
  - Các checkpoint ít thường xuyên hơn sẽ cho phép ghi thuận lợi hơn
  - Các checkpoint ít thường xuyên hơn sẽ tăng thời gian phục hồi

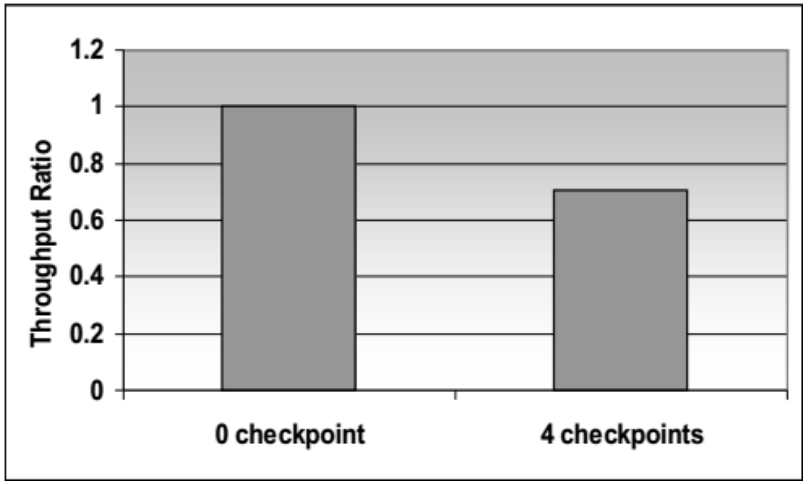
Tối ưu checkpoint trong PostgreSQL (Checkpoint Tuning in PostgreSQL)

- Checkpoint có chi phí:
  - Hoạt động ổ đĩa để chuyển đổi các dirty page thành file dữ liệu
  - Nếu full_page_writes được bật (tránh ghi đĩa không hoàn chỉnh), sau checkpoint một tiền ảnh phải được lưu trữ trong log cho mỗi trang mới bị sửa đổi
- Checkpoint được kích hoạt nếu một trong những nhân tố sau đạt được:
  - checkpoint_timeout (5min): Khoảng thời gian tối đa giữa các checkpoint
  - checkpoint_segments (3): Số lượng lớn nhất của các đoạn file log (16MB)

Tối ưu checkpoint trong PostgreSQL (Checkpoint Tuning in PostgreSQL)

- Mởi rộng băng thông checkpoint (Spreading checkpoint traffic):
  - Băng thông checkpoint được phân tán để giảm việc tải I/O
  - Checkpoint_completion_target (0.5): Khoảng thời gian nhỏ trước khi checkpoint tiếp theo xảy ra
  - Checkpoint nên kết thúc trong khoản thời gian này
- Giám sát checkpoint (Monitoring checkpoints):
  - checkpoint_warning (30s): Ghi cảnh báo đến log nếu các checkpoint xảy ra thường xuyên hơn
  - Xuất hiện thường xuyên thể hiện checkpoint_segments nên được tăng

Tối ưu checkpoint – Thực nghiệm



- Giao dịch dài với nhiều cập nhật
- Các checkpoint kích hoạt khi giao dịch vẫn hoạt động (file log nhỏ hơn)
- Tác động tiêu cực lên hiệu suất:kích thước của các file log nên được tăng.