CHƯƠNG I
GIỚI THIỆU
(Introduction)

MỤC ĐÍCH

Chương này trình bày một cái nhìn bao quát về cơ sở dữ liệu (CSDL/DB), về hệ quản trị cơ sở dữ liệu (HQTCSDL/DBMS) và về hệ cơ sở dữ liệu (HCSDL/DBS). Các đòi hỏi khi xây dựng một HQTCSDL đó cũng chính là những chức năng mà một HCSDL cần phải có. Một khái niệm quan trọng là khái niệm giao dịch (Transaction). Các tính chất một giao dịch phải có để đảm bảo một HQTCSDL, được xây dựng trên HCSDL tương ứng, trong suốt quá trình hoạt động sẽ luôn cho một CSDL tin cậy (dữ liệu luôn nhất quán). Quản trị giao dịch nhằm đảm bảo mọi giao dịch trong hệ thống có các tính chất mà một giao dịch phải có. Một điều cần chú ý là trong các tính chất của một giao dịch, *tính chất nhất quán* trước hết phải được đảm bảo bởi người lập trình-người viết ra giao dịch.

YÊU CẦU

Hiểu các khái niệm.

Hiểu các vấn đề đặt ra khi xây dựng một HQTCSDL: thiết kế CSDL, đảm bảo tính nhất quán của CSDL trong suốt cuộc sống của nó, nền tảng phần cứng trên đó một HQTCSDL được xây dựng.

Hiểu cấu trúc hệ thống tổng thể

Hiểu vai trò của các người sử dụng hệ thống.

MỘT SỐ KHÁI NIỆM

- Một cơ sở dữ liệu (CSDL/ DB: DataBase) là một tập hợp các tập tin có liên quan với nhau, được thiết kế nhằm làm giảm thiểu sự lặp lại dữ liệu.
- Một hệ quản trị cơ sở dữ liệu (HQTCSDL/ DBMS: DataBase Management System) là một hệ thống gồm một CSDL và các thao tác trên CSDL đó, được thiết kế trên một nền tảng phần cứng, phần mềm và với một kiến trúc nhất định.
- Một hệ cơ sở dữ liệu (HCSDL/ DBS: DataBase System) là một phần mềm cho phép xây dựng một HQTCSDL.

HỆ CƠ SỞ DỮ LIỆU

Một số điểm bất lợi chính của việc lưu giữ *thông tin có tổ chức* trong hệ thống xử lý file thông thường:

- **Dư thừa dữ liệu và tính không nhất quán** (Data redundancy and inconsistency): Do các file và các trình ứng dụng được tạo ra bởi các người lập trình khác nhau, nên các file có định dạng khác nhau, các chương trình được viết trong các ngôn ngữ lập trình khác nhau, cùng một thông tin có thể được lưu giữ trong các file khác nhau. Tính không thống nhất và dư thừa này sẽ làm *tăng chi phí truy xuất và lưu trữ*, hơn nũa, nó sẽ dẫn đến tính không nhất quán của dữ liệu: *các bản sao của cùng một dữ liệu có thể không nhất quán*.
- **Khó khăn trong việc truy xuất dữ liệu**: Môi trường của hệ thống xử lý file thông thường không cung cấp các công cụ cho phép truy xuất thông tin một cách hiệu quả và thuận lợi.
- **Sự cô lập dữ liệu** (Data isolation): Các giá trị dữ liệu được lưu trữ trong cơ sở dữ liệu phải thoả mãn một số các *ràng buộc về tính nhất quán của dữ liệu (ràng buộc nhất quán/consistency contraints)*. Trong hệ thống xử lý file thông thường, rất khó khăn trong việc thay đổi các chương trình để thoả mãn các yêu cầu thay đổi ràng buộc. Vấn đề trở nên khó khăn hơn khi các ràng buộc liên quan đến các hạng mục dữ liệu nằm trong các file khác nhau.
- **Các vấn đề về tính nguyên tử** (Atomicity problems): Tính nguyên tử của một hoạt động (giao dịch) là: *hoặc nó được hoàn tất trọn vẹn hoặc không có gì cả*. Điều này có nghĩa là một hoạt động (giao dịch) *chỉ làm thay đổi* các dữ liệu bền vững khi nó đã hoàn tất (kết thúc thành công) nếu không, giao dịch không để lại một dấu vết nào trên CSDL. Trong hệ thống xử lý file thông thường khó đảm bảo được tính chất này.
- **Tính bất thường trong truy xuất cạnh tranh**: Một hệ thống cho phép nhiều người sử dụng cập nhật dữ liệu đồng thời, có thể dẫn đến kết quả là dữ liệu không nhất quán. Điều này đòi hỏi một sự giám sát. Hệ thống xử lý file thông thường không cung cấp chức năng này.
- **Vấn đề an toàn** (Security problems): một người sử dụng hệ cơ sở dữ liệu không cần thiết và cũng không có quyền truy xuất tất cả các dữ liệu. Vấn đề này đòi hỏi hệ thống phải đảm bảo được tính phân quyền, chống truy xuất trái phép ...

Các bất lợi nêu trên đã gợi mở sự phát triển các DBMS. Phần sau của giáo trình sẽ đề cập đến các quan niệm và các thuật toán được sử dụng để phát triển một hệ cơ sở dữ liệu nhằm *giải quyết các vấn đề nêu trên*. Một số khái niệm

GÓC NHÌN DỮ LIỆU

Tính hiệu quả của hệ thống đòi hỏi phải thiết kế các cấu trúc dữ liệu phức tạp để biểu diễn dữ liệu trong cơ sở dữ liệu. Các nhà phát triển che dấu sự phức tạp này thông qua các *mức trừu tượng* nhằm đơn giản hóa sự trao đổi của người sử dụng với hệ thống:

- **Mức vật lý (Physical level)**: Mức thấp nhất của sự trừu tượng, mô tả dữ liệu hiện được lưu trữ thế nào. Ở mức này, cấu trúc dữ liệu mức thấp, phức tạp được mô tả chi tiết.
- **Mức luận lý (Logical level)**: Mức kế cao hơn về sự trừu tượng, mô tả dữ liệu gì được lưu trữ trong cơ sở dữ liệu và các mối quan hệ gì giữa các dữ liệu này. Mức logic của sự trừu tượng được dùng bởi các người quản trị cơ sở dữ liệu.
- **Mức view (view level)**: Mức cao nhất của sự trừu tượng, mô tả chỉ một phần của cơ sở dữ liệu toàn thể. Một người sử dụng cơ sở dữ liệu liên quan đến chỉ một bộ phận của cơ sở dữ liệu. Như vậy sự trao đổi của họ với hệ thống được làm đơn giản bởi việc định nghĩa view. Hệ thống có thể cung cấp nhiều mức view đối với cùng một cơ sở dữ liệu.

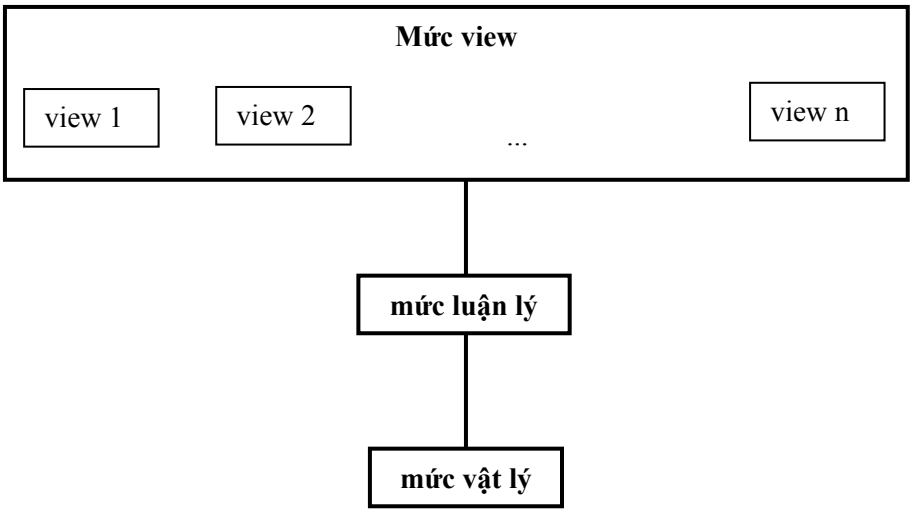


Figure 1

- **Thể hiện và sơ đồ** (instances and schemas): Tập hợp các thông tin được lưu trữ trong cơ sở dữ liệu tại một thời điểm được gọi là một thể hiện (instance) của cơ sở dữ liệu. Thiết kế tổng thể của cơ sở dữ liệu được gọi là sơ đồ (schema).

Một hệ cơ sở dữ liệu có một vài sơ đồ, được phân tương ứng với các mức trừu tượng. ở mức thấp nhất là **sơ đồ vật lý** (physical schema), ở mức trung gian là **sơ đồ luận lý** (logical schema), ở mức cao nhất là **sơ đồ con** (subschema). Nói chung một hệ cơ sở dữ liệu hỗ trợ một sơ đồ vật lý, một sơ đồ luận lý và một vài sơ đồ con.

- Khả năng sửa đổi một định nghĩa ở một mức không ảnh hưởng một định nghĩa sơ đồ ở mức cao hơn được gọi là **sự độc lập dữ liệu** (data independence). Có hai mức độc lập dữ liệu:
  - **Độc lập dữ liệu vật lý** (Physical data independence) là khả năng sửa đổi sơ đồ vật lý không làm cho các chương trình ứng dụng phải viết lại. Các sửa đổi ở mức vật lý là cần thiết để cải thiện hiệu năng.

- **Độc lập dữ liệu luận lý** (Logical data independence) là khả năng sửa đổi sơ đồ luận lý không làm cho các chương trình ứng dụng phải viết lại. Các sửa đổi ở mức luận lý là cần thiết khi cấu trúc luận lý của cơ sở dữ liệu bị thay thế.

MÔ HÌNH DỮ LIỆU

Nằm dưới cấu trúc của một cơ sở dữ liệu là mô hình dữ liệu: một bộ các công cụ quan niệm để mô tả dữ liệu, quan hệ dữ liệu, ngữ nghĩa dữ liệu và các ràng buộc nhất quán. Có ba nhóm mô hình: Các mô hình luận lý dựa trên đối tượng (Object-based logical models), các mô hình luận lý dựa trên mẩu tin (record-based logical models), các mô hình vật lý (physical models).

- **Các mô hình luận lý dựa trên đối tượng** được dùng mô tả dữ liệu ở mức luận lý và mức view. Chúng được đặc trưng bởi việc chúng cung cấp khả năng cấu trúc linh hoạt và cho phép các ràng buộc dữ liệu được xác định một cách tường minh. Dưới đây là một vài mô hình được biết rộng rãi: Mô hình **thực thể - quan hệ** (entity-relationship model), mô hình **hướng đối tượng** (object-oriented model), mô hình **dữ liệu ngữ nghĩa** (semantic data model), mô hình **dữ liệu hàm** (function data model).
- **Các mô hình luận lý dựa trên mẩu tin** được dùng để miêu tả dữ liệu ở mức luận lý hay mức view. Chúng được dùng để xác định cấu trúc luận lý toàn thể của cơ sở dữ liệu và cung cấp sự mô tả mức cao hơn việc thực hiện. Cơ sở dữ liệu được cấu trúc ở dạng mẩu tin định dạng cố định (fixed format record): mỗi mẩu tin xác định một số cố định các trường, mối trường thường có độ dài cố định. Một vài mô hình được biết rộng rãi là: **Mô hình quan hệ, mô hình mạng, mô hình phân cấp**.
- **Mô hình dữ liệu vật lý** được dùng để mô tả dữ liệu ở mức thấp nhất. Hai mô hình dữ liệu vật lý được biết rộng rãi nhất là **mô hình hợp nhất** (unifying model) và **mô hình khung-bộ nhớ** (frame-memory model).

NGÔN NGỮ CƠ SỞ DỮ LIỆU

Một hệ cơ sở dữ liệu cung cấp hai kiểu ngôn ngữ khác nhau: một để xác định sơ đồ cơ sở dữ liệu, một để biểu diễn các vấn tin cơ sở dữ liệu và cập nhật.

- **Ngôn ngữ định nghĩa dữ liệu (Data Definition Language: DDL)** cho phép định nghĩa sơ đồ cơ sở dữ liệu. Kết quả biên dịch các lệnh của DDL là tập hợp các bảng được lưu trữ trong một ***file đặc biệt được gọi là tự điển dữ liệu (data dictionary)*** hay thư mục dữ liệu (***data directory***). Tự điển dữ liệu là một file chứa ***metadata***. File này được tra cứu trước khi dữ liệu hiện hành được đọc hay sửa đổi. Cấu trúc lưu trữ và phương pháp truy cập được sử dụng bởi hệ cơ sở dữ liệu được xác định bởi một tập hợp các định nghĩa trong một kiểu đặc biệt của DDL được gọi là ***ngôn ngữ định nghĩa và lưu trữ dữ liệu (data storage and definition language)***. Kết quả biên dịch của các định nghĩa này là một tập hợp các chỉ thị xác định sự thực hiện chi tiết của các sơ đồ cơ sở dữ liệu (thường được che dấu).
- **Ngôn ngữ thao tác dữ liệu (Data manipulation language: DML)** là ngôn ngữ cho phép người sử dụng truy xuất hoặc thao tác dữ liệu. Có hai kiểu ngôn ngữ thao tác dữ liệu: ***DML thủ tục (procedural DML)*** yêu cầu người sử dụng đặc tả dữ liệu nào cần và làm thế nào để nhận được nó. ***DML không thủ tục (Nonprocedural DML)*** yêu cầu người sử dụng đặc tả dữ liệu nào cần nhưng không cần đặc tả làm thế nào để nhận được nó. Một vấn tin (***query***) là một lệnh yêu cầu tìm lại dữ liệu (***information***

retrieval). Phần ngôn ngữ DML liên quan đến sự tìm lại thông tin được gọi là ngôn ngữ vấn tin (***query language***).

QUẢN TRỊ GIAO DỊCH

Thông thường, một số thao tác trên cơ sở dữ liệu tạo thành một đơn vị logic công việc. Ta hãy xét ví dụ chuyển khoản, trong đó một số tiền x được chuyển từ tài khoản A (A:=A-x) sang một tài khoản B (B:=B+x). Một yếu tố cần thiết là cả hai thao tác này hoặc cùng xảy ra hoặc không hoạt động nào xảy ra cả. Việc chuyển khoản phải xảy ra trong tính toàn thể của nó hoặc không. Đòi hỏi ***toàn thể-hoặc-không*** này được gọi là tính nguyên tử (atomicity). Một yếu tố cần thiết khác là sự thực hiện việc chuyển khoản bảo tồn tính nhất quán của cơ sở dữ liệu: *giá trị của tổng A + B phải được bảo tồn*. Đòi hỏi về tính chính xác này được gọi là tính nhất quán (consistency). Cuối cùng, sau khi thực hiện thành công hoạt động chuyển khoản, các giá trị của các tài khoản A và B phải bền vững cho dù có thể có sự cố hệ thống. Đòi hỏi về tính bền vững này được gọi là tính lâu bền (durability).

Một giao dịch là một tập các hoạt động thực hiện chỉ một chức năng logic trong một ứng dụng cơ sở dữ liệu. Mỗi giao dịch là một đơn vị mang cả tính nguyên tử lẫn tính nhất quán. Như vậy, các giao dịch phải không được vi phạm bất kỳ ràng buộc nhất quán nào: ***Nếu cơ sở dữ liệu là nhất quán khi một giao dịch khởi động thì nó cũng phải là nhất quán khi giao dịch kết thúc thành công***. Tuy nhiên, trong khi đang thực hiện giao dịch, phải cho phép sự không nhất quán tạm thời. Sự không nhất quán tạm thời này tuy là cần thiết nhưng lại có thể dẫn đến các khó khăn nếu xảy ra sự cố.

Trách nhiệm của người lập trình là xác định đúng đắn các giao dịch sao cho mỗi một bảo tồn tính nhất quán của cơ sở dữ liệu.

Đảm bảo tính nguyên tử và tính lâu bền là trách nhiệm của hệ cơ sở dữ liệu nói chung và của ***thành phần quản trị giao dịch (transaction-management component)*** nói riêng. Nếu không có sự cố, tất cả giao dịch hoàn tất thành công và tính nguyên tử được hoàn thành dễ dàng. Tuy nhiên, do sự hiện diện của các sự cố, một giao dich có thể không hoàn tất thành công sự thực hiện của nó. Nếu tính nguyên tử được đảm bảo, một giao dịch thất bại không gây hiệu quả đến trạng thái của cơ sở dữ liệu. Như vậy, cơ sở dữ liệu phải được hoàn lại trạng thái của nó trước khi giao dịch bắt đầu. Hệ cơ sở dữ liệu phải có trách nhiệm phát hiện sự cố hệ thống và trả lại cơ sở dữ liệu về trạng thái trước khi xảy ra sự cố.

Khi một số giao dịch cạnh tranh cập nhật cơ sở dữ liệu, tính nhất quán của dữ liệu có thể không được bảo tồn, ngay cả khi mỗi giao dịch là chính xác. ***Bộ quản trị điều khiển cạnh tranh (concurency-control manager)*** có trách nhiệm điều khiển các trao đổi giữa các giao dịch cạnh tranh để đảm bảo tính thống nhất của CSDL.

QUẢN TRỊ LƯU TRỮ

Các CSDL đòi hỏi một khối lượng lớn không gian lưu trữ, có thể lên đến nhiều terabytes (1 terabyte=10^3 Gigabytes=10^6 Megabytes). Các thông tin phải được lưu trữ trên lưu trữ ngoài (đĩa). Dữ liệu được di chuyển giữa lưu trữ đĩa và bộ nhớ chính khi cần thiết. Do việc di chuyển dữ liệu từ và lên đĩa tương đối chậm so với tốc độ của đơn vị xử lý trung tâm, điều này ép buộc hệ CSDL phải cấu trúc dữ liệu sao cho tối ưu hóa nhu cầu di chuyển dữ liệu giữa đĩa và bộ nhớ chính.

Mục đích của một hệ CSDL là làm đơn giản và dễ dàng việc truy xuất dữ liệu. Người sử dụng hệ thống có thể không cần quan tâm đến chi tiết vật lý của sự thực thi hệ thống. Phần lớn họ chỉ quan đến hiệu năng của hệ thống (thời gian trả lời một câu vấn tin ...).

Bộ quản trị lưu trữ (storage manager) là một module chương trình cung cấp giao diện giữa dữ liệu mức thấp được lưu trữ trong CSDL với các chương trình ứng dụng và các câu vấn tin được đệ trình cho hệ thống. Bộ quản trị lưu trữ có trách nhiệm trao đổi với bộ quản trị file (file manager). Dữ liệu thô được lưu trữ trên đĩa sử dụng hệ thống file (file system), hệ thống này thường được cung cấp bởi hệ điều hành. Bộ quản trị lưu trữ dịch các câu lệnh DML thành các lệnh của hệ thống file mức thấp. Như vậy, *bộ quản trị lưu trữ có nhiệm vụ lưu trữ, tìm lại và cập nhật dữ liệu trong CSDL*.

NHÀ QUẢN TRỊ CƠ SỞ DỮ LIỆU

Một trong các lý do chính đối với việc sử dụng DBMS là có sự điều khiển trung tâm cho cả dữ liệu lẫn các chương trình truy cập các dữ liệu này. Người điều khiển trung tâm trên toàn hệ thống như vậy gọi là nhà quản trị cơ sở dữ liệu (DataBase Administrator - DBA). Các chức năng của DBA như sau:

- ***Định nghĩa sơ đồ***: DBA tạo ra sơ đồ CSDL gốc bằng cách viết một tập các định nghĩa mà nó sẽ được dịch bởi trình biên dịch DDL thành một tập các bảng được lưu trữ thường trực trong tự điển dữ liệu.
- ***Định nghĩa cấu trúc lưu trữ và phương pháp truy xuất***: DBA tạo ra một cấu trúc lưu trữ thích hợp và các phương pháp truy xuất bằng cách viết một tập hợp các định nghĩa mà nó sẽ được dịch bởi trình biên dịch lưu trữ dữ liệu và ngôn ngữ định nghĩa dữ liệu.
- ***Sửa đổi sơ đồ và tổ chức vật lý***
- ***Cấp quyền truy xuất dữ liệu***: Việc cấp các dạng quyền truy cập khác nhau cho phép DBA điều hoà những phần của CSDL mà nhiều người có thể truy xuất. Thông tin về quyền được lưu giữ trong một cấu trúc hệ thống đặc biệt, nó được tham khảo bởi hệ CSDL mỗi khi có sự truy xuất dữ liệu của hệ thống.
- ***Đặc tả ràng buộc toàn vẹn (integrity-contraint)***: Các giá trị dữ liệu được lưu trữ trong CSDL phải thoả mãn một số các ràng buộc nhất quán nhất định. Ví dụ số giờ làm việc của một nhân viên trong một tuần không thể vượt quá một giới hạn 80 giờ chẳng hạn. Một ràng buộc như vậy phải được đặc tả một cách tường minh bởi DBA. Các ràng buộc toàn vẹn được lưu giữ trong một cấu trúc hệ thống đặc biệt được tham khảo bởi hệ CSDL mỗi khi có sự cập nhật dữ liệu.

NGƯỜI SỬ DỤNG CSDL

Mục đích đầu tiên của hệ CSDL là cung cấp một môi trường để tìm lại thông tin và lưu thông tin trong CSDL. Các người sử dụng cơ sở dữ liệu được phân thành bốn nhóm tuỳ theo cách thức họ trao đổi với hệ thống.

- ***Các người lập trình ứng dụng***: Là nhà chuyên môn máy tính người trao đổi với hệ thống thông qua các lời gọi DML được nhúng trong một chương trình được viết trong một ngôn ngữ chủ - *host language* (Pascal, C, Cobol ...). Các chương trình này thường được tham khảo như các chương trình ứng dụng. Vì cú pháp DML thường rất khác với cú pháp của ngôn ngữ chủ, các lời gọi DML thường được bắt đầu bởi một ký tự đặc biệt như vậy mã thích hợp mới có thể được sinh. Một bộ tiền xử lý đặc biệt, được gọi là tiền

biên dịch (precompiler) DML, chuyển các lệnh DML thành các lời gọi thủ tục chuẩn trong ngôn ngữ chủ. Bộ biên dịch ngôn ngữ chủ sẽ sinh mã đối tượng thích hợp. Có những ngôn ngữ lập trình phối hợp cấu trúc điều khiển của các ngôn ngữ giống như Pascal với cấu trúc điều khiển để thao tác đối tượng CSDL. Các ngôn ngữ này (đôi khi được gọi là ngôn ngữ thế hệ thứ tư) thường bao gồm các đặc điểm đặc biệt để làm dễ dàng việc sinh các dạng và hiển thị dữ liệu trên màn hình.

- ***Các người sử dụng thành thạo (Sophisticated users)***: Trao đổi với hệ thống không qua viết trình. Thay vào đó họ đặt ra các yêu cầu của họ trong ngôn ngữ truy vấn CSDL (Database query language). Mỗi câu vấn tin như vậy được đệ trình cho bộ xử lý vấn tin, chức năng của bộ xử lý vấn tin là "dịch" các lệnh DML thành các chỉ thị mà bộ quản trị lưu trữ hiểu. Các nhà phân tích đệ trình các câu vấn tin thăm dò dữ liệu trong cơ sở dữ liệu thuộc vào phạm trù này.
- ***Các người sử dụng chuyên biệt (Specialized users)***: Là các người sử dụng thành thạo, họ viết các ứng dụng CSDL chuyên biệt không nằm trong khung xử lý dữ liệu truyền thống. Trong đó, phải kể đến các hệ thống thiết kế được trợ giúp bởi máy tính (computer-aided design systems), Cơ sở tri thức (knowledge-base) và hệ chuyên gia (expert systems), các hệ thống lưu trữ dữ liệu với kiểu dữ liệu phức tạp (dữ liệu đồ họa, hình ảnh, âm thanh) và các hệ thống mô hình môi trường (environment-modeling systems)
- ***Các người sử dụng ngây thơ (Naive users)***: là các người sử dụng không thành thạo, họ trao đổi với hệ thống bởi cầu dẫn một trong các chương trình ứng dụng thường trực đã được viết sẵn.

CẤU TRÚC HỆ THỐNG TỔNG THỂ

Một hệ CSDL được phân thành các module, mỗi một thực hiện một trách nhiệm trong hệ thống tổng thể. Một số chức năng của hệ CSDL có thể được cung cấp bởi hệ điều hành. Trong hầu hết các trường hợp, hệ điều hành chỉ cung cấp các dịch vụ cơ sở nhất, hệ CSDL phải xây dựng trên cơ sở đó. Như vậy, thiết kế hệ CSDL phải xem xét đến giao diện giữa hệ CSDL và hệ điều hành.

Các thành phần chức năng của hệ CSDL có thể được chia thành các thành phần xử lý vấn tin (query processor components) và các thành phần quản trị lưu trữ (storage manager components).

Các thành phần xử lý vấn tin gồm:

- ***Trình biên dịch DML (DML compiler):*** dịch các lệnh DML trong một ngôn ngữ vấn tin thành các chỉ thị mức thấp mà engine định giá vấn tin (query evaluation engine) có thể hiểu. Hơn nữa, Trình biên dịch DML phải biến đổi một yêu cầu của người sử dụng thành một đích tương đương nhưng ở dạng hiệu quả hơn có nghĩa là tìm một chiến lược tốt để thực hiện câu vấn tin.
- ***Trình tiền biên dịch DML nhúng (Embedded DML Precompiler):*** biến đổi các lệnh DML được nhúng trong một chương trình ứng dụng thành các lời gọi thủ tục chuẩn trong ngôn ngữ chủ. Trình tiền biên dịch phải trao đổi với trình biên dịch DML để sinh mã thích hợp.
- ***Bộ thông dịch DDL (DDL interpreter):*** thông dịch các lệnh DDL và ghi chúng vào một tập hợp các bảng chứa metadata.

- ***Engine định giá vấn tin (Query evaluation engine):*** Thực hiện các chỉ thị mức thấp được sinh ra bởi trình biên dịch DML.

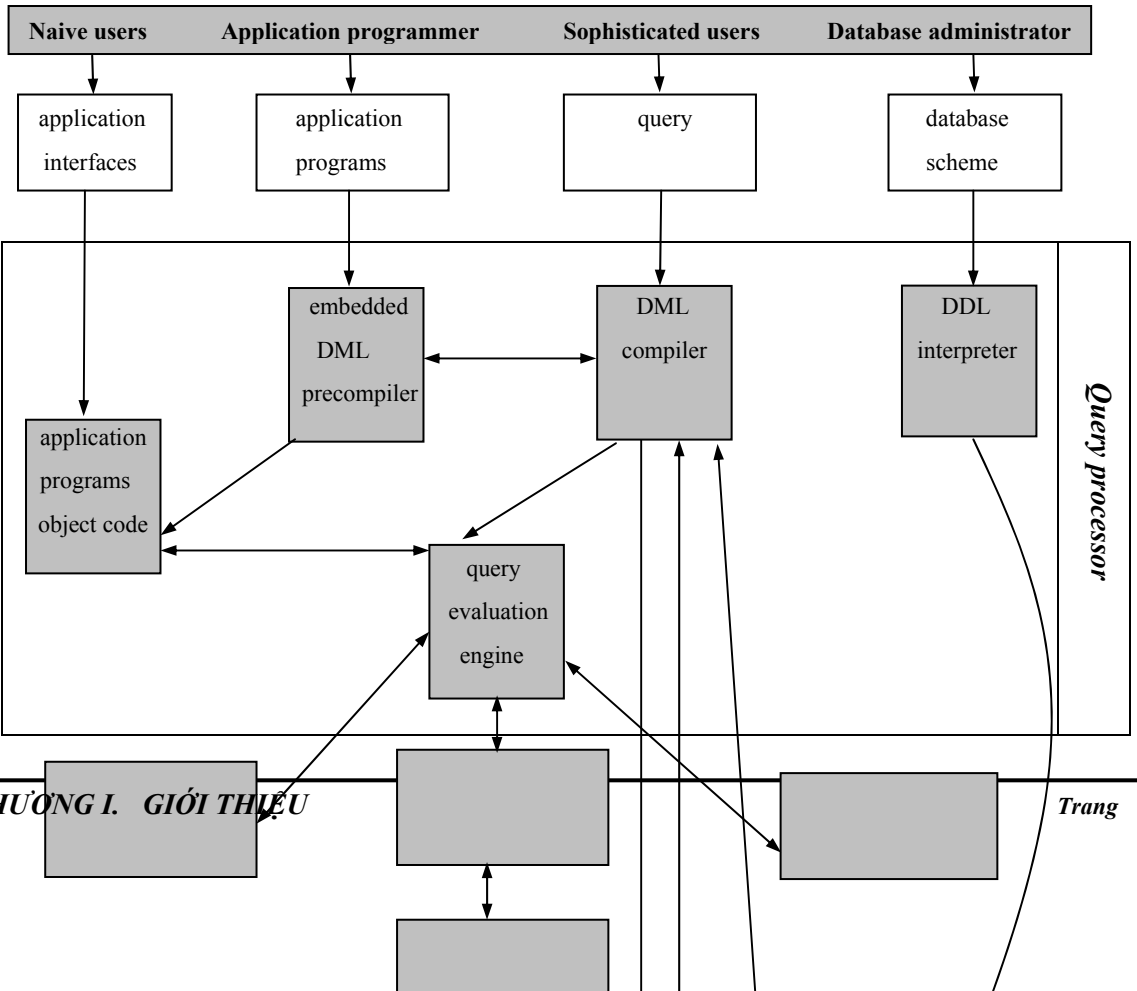
Các thành phần quản trị lưu trữ cung cấp các giao diện giữa dữ liệu mức thấp được lưu trữ trong CSDL và các chương trình ứng dụng, các vấn tin được đệ trình cho hệ thống. Các thành phần quản trị lưu trữ gồm:

- ***Bộ quản trị quyền và tính toàn vẹn (Authorization and integrity manager):*** kiểm tra sự thoả mãn các ràng buộc toàn vẹn và kiểm tra quyền truy xuất dữ liệu của người sử dụng.
- ***Bộ quản trị giao dịch (Transaction manager):*** Đảm bảo rằng CSDL được duy trì trong trạng thái nhất quán cho dù hệ thống có sự cố và đảm bảo rằng các thực hiện giao dịch cạnh tranh tiến triển không xung đột.
- ***Bộ quản trị file (File manager):*** Quản trị cấp phát không gian trên lưu trữ đĩa và các cấu trúc dữ liệu được dùng để biểu diễn thông tin được lưu trữ trên đĩa.
- ***Bộ quản trị bộ đệm (Buffer manager):*** có trách nhiệm đem dữ liệu từ lưu trữ đĩa vào bộ nhớ chính và quyết định dữ liệu nào trữ trong bộ nhớ.

Hơn nữa, một số cấu trúc dữ liệu được cần đến như bộ phận của sự thực thi hệ thống vật lý:

- ***Các file dữ liệu:*** Lưu trữ CSDL
- ***Tự điển dữ liệu (Data Dictionary):*** lưu metadata về cấu trúc CSDL.
- ***Chỉ mục (Indices):*** cung cấp truy xuất nhanh đến các hạng mục dữ liệu chứa các giá trị tìm kiếm.
- ***Dữ liệu thống kê (Statistical data):*** lưu trữ thông tin thống kê về dữ liệu trong cơ sở dữ liệu. Thông tin này được dùng bởi bộ xử lý vấn tin để chọn những phương pháp hiệu quả thực hiện câu vấn tin.

Sơ đồ các thành phần và các nối kết giữa chúng



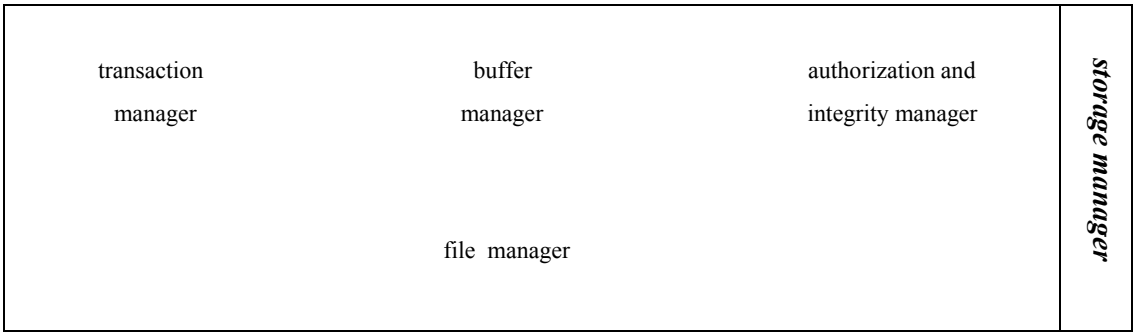


Figure 2

KIẾN TRÚC HỆ CƠ SỞ DỮ LIỆU

Kiến trúc hệ CSDL bị ảnh hưởng nhiều bởi hệ thống máy nền. Các sắc thái của kiến trúc máy như mạng, song song và phân tán được phản ánh trong kiến trúc của hệ CSDL.

- Mạng máy tính cho phép thực hiện một số công việc trên một hệ thống các server, một số công việc trên các hệ thống client. Việc phân chia công việc này dẫn đến sự phát triển hệ CSDL *client-server.*
- Xử lý song song trong một hệ thống máy tính làm tăng tốc độ các hoạt động của hệ CSDL, trả lời các giao dịch nhanh hơn. Các vấn tin được xử lý theo cách khai thác tính song song. Sự cần thiết xử lý vấn tin song song này dẫn tới sự phát triển của hệ CSDL *song song*.
- Dữ liệu phân tán trên các site hoặc trên các bộ phận trong một cơ quan cho phép các dữ liệu thường trú tại nơi chúng được sinh ra nhưng vẫn có thể truy xuất chúng từ các site khác hay các bộ phận khác. Việc lưu nhiều bản sao của CSDL trên các site khác nhau cho phép các tổ chức lớn vẫn có thể tiếp tục hoạt động khi một hay một vài site bị sự cố. Hệ CSDL phân tán được phát triển để quản lý dữ liệu phân tán, trên phương diện địa lý hay quản trị, trải rộng trên nhiều hệ CSDL .

HỆ THỐNG TẬP TRUNG

Các hệ CSDL tập trung chạy trên máy đơn và không trao đổi với các máy khác. Các hệ thống như vậy trải từ các hệ CSDL một người sử dụng chạy trên các máy cá nhân (PC) đến các hệ CSDL hiệu năng cao chạy trên các hệ mainframe. Một hệ máy tính mục đích chung hiện đại gồm một hoặc một vài CPU và một số bộ điều khiển thiết bị được nối với nhau thông qua một bus

chung, cho phép truy xuất đến bộ nhớ chia sẻ. CPU có bộ nhớ cache cục bộ lưu các bản sao của một số phần của bộ nhớ chính nhằm tăng tốc độ truy xuất dữ liệu. Mỗi bộ điều khiển thiết bị phụ trách một kiểu thiết bị xác định. Các CPU và các bộ điều khiển thiết bị có thể thực hiện đồng thời, canh tranh truy cập bộ nhớ. Bộ nhớ cache giúp làm giảm sự tranh chấp truy xuất bộ nhớ. Ta phân biệt hai cách các máy tính được sử dụng: Hệ thống một người dùng và hệ thống nhiều người dùng. Hệ CSDL được thiết kế cho hệ thống một người dùng không hỗ trợ điều khiển cạnh tranh, chức năng phục hồi hoặc là thiếu hoặc chỉ là một sự chép dự phòng đơn giản.

HỆ THỐNG CLIENT-SERVER

Các máy tính cá nhân (PC) ngày càng trở nên mạnh hơn, nhanh hơn, và rẻ hơn. Có sự chuyển dịch trong hệ thống tập trung. Các đầu cuối (terminal) được nối với hệ thống tập trung bây giờ được thế chỗ bởi các máy tính cá nhân. Chức năng giao diện người dùng (user interface) thường được quản lý trực tiếp bởi các hệ thống tập trung nay được quản lý bởi các máy tính cá nhân. Như vậy, các hệ thống tập trung ngày nay hoạt động như các hệ thống server nó làm thoả mãn các đòi hỏi của các client. Chức năng CSDL có thể được chia thành hai phần: phần trước (front-end) và phần sau (back-end). Phần sau quản trị truy xuất cấu trúc, định giá câu vấn tin và tối ưu hoá, điều khiển sự xảy ra đồng thời và phục hồi. Phần trước của hệ CSDL gồm các công cụ như: tạo mẫu (form), các bộ soạn báo cáo (report writer), giao diện đồ hoạ người dùng (graphical user interface). Giao diện giữa phần trước và phần sau thông qua SQL hoặc một chương trình ứng dụng. Các hệ thống server có thể được phân thành các phạm trù : server giao dịch (transaction server), server dữ liệu (data server).

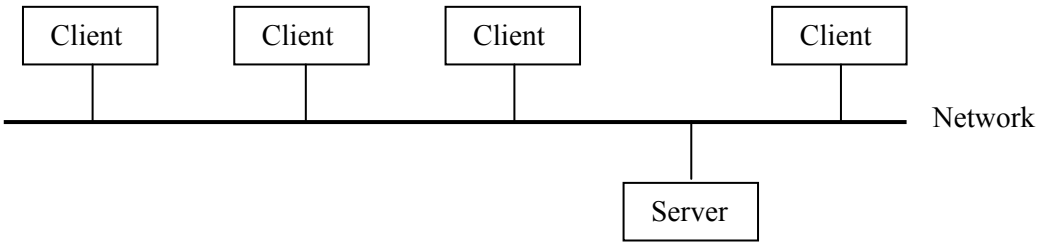


Figure 3

- ***Hệ thống server giao dịch (transaction-server systems)***: còn được gọi là hệ thống server vấn tin (query-server system), cung cấp một giao diện mà các client có thể gửi đến nó các yêu cầu thực hiện một hành động. Để đáp ứng các yêu cầu, hệ thống thực hiện các hành động và gửi lại client các kết quả. Các người sử dụng có thể đặc tả các yêu cầu trong SQL hoặc trong một giao diện trình ứng dụng sử dụng một cơ chế gọi thủ tục xa (remote-procedure-call).
    - ***Các servers giao dịch (Transaction servers):*** Trong các hệ thống tập trung, phần trước (front-end) và phần sau (back-end) được thực hiện trong một hệ thống. Kiến trúc server giao dịch cho phép chia chức năng giữa phần trước và phần sau. Chức năng phần trước được hỗ trợ trên các máy tính cá nhân (PC). Các PC hành động như những khách hàng của các hệ thống server nơi lưu trữ một khối lượng lớn dữ liệu và hỗ trợ các chức năng phần sau. Các clients gửi các giao dịch đến các hệ thống server tại đó các giao dịch được thực hiện và

các kết quả được gửi trả lại cho các clients, người giữ trách nhiệm hiển thị dữ liệu.

ODBC (Open DataBase Connectivity) được phát triển để tạo giao diện giữa các clients và các servers. ODBC là một giao diện trình ứng dụng cho phép các clients sinh ra các lệnh SQL và gửi đến một server tại đó lệnh được thực hiện. Bất kỳ client nào sử dụng giao diện có thể nối với bất kỳ một server nào cung cấp giao diện này.

Các giao diện client-server khác ODBC cũng được sử dụng trong một số hệ thống xử lý giao dịch. Chúng được xác định bởi một giao diện lập trình ứng dụng, sử dụng nó các clients tạo ra các lời gọi thủ tục giao dịch từ xa (transactional remote procedure calls) trên server. Các lời gọi này giống như các lời gọi thủ tục gốc đối với người lập trình nhưng tất cả các lời gọi thủ tục từ xa của một client được bao trong một giao dịch ở server cuối. Như vậy nếu giao dịch bỏ dở, server có thể huỷ bỏ hiệu quả của các lời gọi thủ tục xa riêng lẻ.

- ***Hệ thống server dữ liệu (Data-server systems)***: cho phép các clients trao đổi với các server bằng cách tạo ra các yêu cầu đọc hoặc cập nhật dữ liệu trong các đơn vị như file hoặc trang. Ví dụ, các file-servers cung cấp một giao diện với hệ thống file tại đó các clients có thể tạo, cập nhật, đọc hoặc xoá files. Các servers dữ liệu của cơ sở dữ liệu cung cấp nhiều chức năng hơn; chúng hỗ trợ các đơn vị dữ liệu nhỏ hơn file như trang, bộ (tuple) hoặc đối tượng. Chúng cũng cung cấp phương tiện dễ dàng để lấy chỉ mục (indexing) dữ liệu, phương tiện dễ dàng để tạo giao dịch.
    - ***Các server dữ liệu (Data Servers):*** Các hệ thống server dữ liệu được sử dụng trong các mạng cục bộ, trong đó có một nối kết tốc độ cao giữa các máy clients và máy server, các máy clients có sức mạnh xử lý tương thích với máy server và các công việc phải được thực hiện là tăng cường tính toán. Trong một môi trường như vậy, có thể gửi dữ liệu đến các máy client để thực hiện tất cả các xử lý tại máy clients sau đó gửi dữ liệu trở lại đến máy server. Kiến trúc này đòi hỏi các tính năng back-end đầy đủ tại các clients. Kiến trúc server dữ liệu thường được gặp trong các hệ CSDL hướng đối tượng (Object-Oriented DataBase Systems)

      Gửi trang đối lại với gửi hạng mục (Page shipping versus item shipping): Đơn vị liên lạc dữ liệu có thể là các "hạt thô" (Coarse granularity) như một trang, hay hạt min (fine granularity) như một bộ (tuple)/ đối tượng (object). Ta dùng thuật ngữ hạng mục để chỉ bộ hay đối tượng. Nếu đơn vị liên lạc là một hạng mục sẽ dẫn đến tổng chi phí truyền thông điệp tăng. Đem về hạng mục (fetching item) trước khi nó được yêu cầu, được gọi là đem về trước (Prefetching). Gửi trang có thể được xem như một dạng của đem về trước nếu một trang chứa nhiều hạng mục.

      Chốt (Locking): Các chốt thường được cấp bởi server trên các hạng mục mà nó gửi cho các máy clients. *Khi client giữ một chốt trên một hạng mục dữ liệu, nó có quyền "sử dụng" hạng mục dữ liệu này, hơn nữa trong khoảng thời gian client giữ chốt trên hạng mục dữ liệu không một client nào khác có thể sử dụng hạng mục dữ liệu này*. Bất lợi của gửi trang là các máy client có thể được cấp các chốt "hạt quá thô" -- một chốt trên một trang ẩn chứa các chốt trên tất cả các hạng mục trong trang. Các kỹ thuật nhằm tiết giảm chốt (lock deescalation) được đề nghị, trong đó server có thể yêu cầu các clients truyền trả lại các chốt

trên các hạng mục cấp phát trước. Nếu máy client không cần hạng mục cấp phát trước, nó có thể truyền trả lại các chốt trên hạng mục cho server và các chốt này có thể được cấp phát cho các clients khác.

Trữ dữ liệu (Data caching): Dữ liệu được gửi đến một client với danh nghĩa một giao dịch có thể được trữ ở client, ngay cả khi giao dịch đã hoàn tất, nếu không gian lưu trữ có sẵn. Các giao dịch liên tiếp tại cùng một client có thể dùng dữ liệu được trữ. Tuy nhiên, sự kết dính dữ liệu là một vấn đề cần phải được xem xét: một giao dịch tìm thấy dữ liệu được trữ, nó phải chắc chắn rằng dữ liệu này là "mới nhất" vì các dữ liệu này có thể được cập nhật bởi một client khác sau khi chúng được trữ. Như vậy, vẫn phải trao đổi với server để kiểm tra tính hợp lệ của dữ liệu và để giành được một chốt trên dữ liệu.

Trữ chốt (Lock caching): Các chốt cũng có thể được trữ lại tại máy client. Nếu một hạng mục dữ liệu được tìm thấy trong cache và chốt yêu cầu cho một truy xuất đến hạng mục dữ liệu này cũng tìm thấy trong cache, thì việc truy xuất có thể tiến hành không cần một liên lạc nào với server. Tuy nhiên, server cũng phải lưu lại vết của các chốt được trữ. Nếu một client đòi hỏi một chốt từ server, server phải gọi lại tất cả các chốt xung đột trên cùng hạng mục dữ liệu từ tất cả các máy clients đã trữ các chốt.

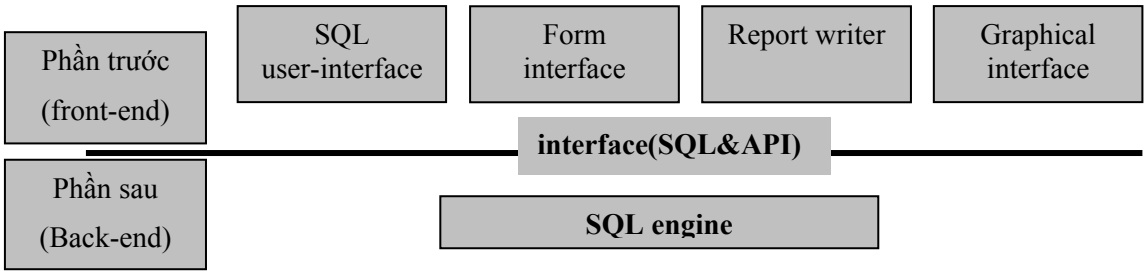


Figure 4

CÁC HỆ SONG SONG (Parallel Systems):

Các hệ song song cải tiến tốc độ xử lý và tốc độ I/O bằng cách sử dụng nhiều CPU và nhiều đĩa song song. Trong xử lý song song, nhiều hoạt động được thực hiện đồng thời. Một máy song song "hạt thô" (coarse-grain) gồm một số nhỏ các bộ xử lý mạnh. Một máy song song đồ sộ (massively parallel) hay "hạt mịn" (fine-grain) sử dụng hàng ngàn bộ xử lý nhỏ hơn. Có hai biện pháp chính để đánh giá hiệu năng của một hệ CSDL. Thứ nhất là năng lực truyền qua (throughput): *số công việc có thể được hoàn tất trong một khoảng thời gian đã cho*. Thứ hai là thời gian đáp ứng (response time): *lượng thời gian cần thiết để hoàn thành một công việc từ lúc nó được đệ trình*. Một hệ thống xử lý một lượng lớn các giao dịch nhỏ có thể cải tiến năng lực truyền qua bởi xử lý song song nhiều giao dịch. Một hệ thống xử lý các giao dịch lớn có thể cải tiến thời gian đáp ứng cũng như năng lực truyền qua bởi thực hiện song song các công việc con (subtask) của mỗi giao dịch.

- **Tăng tốc độ và tăng quy mô (Speedup & Scaleup)**: Tăng tốc độ ám chỉ việc chạy một công việc đã cho trong thời gian ngắn hơn bằng cách tăng bậc song song. Tăng quy mô ám chỉ việc quản lý các công việc lớn bằng cách tăng bậc song song. Chúng ta hãy xét một ứng dụng CSDL chạy trên một hệ thống song song với một số processor và một số đĩa. Giả sử, chúng ta tăng kích cỡ của hệ thống bằng cách tăng số processor, đĩa, và các thành phần khác của hệ thống. Mục đích là xử lý công việc trong thời gian tỷ lệ nghịch với số processor và đĩa được cấp phát.

Giả sử, thời gian thực hiện một công việc trên một máy tính lớn là $\mathbf{T_L}$, thời gian thực hiện cùng công việc này trên máy tính nhỏ là $\mathbf{T_S}$. Tăng tốc độ nhờ song song được định nghĩa là tỷ số $\mathbf{T_S/T_L}$, hệ thống song song được gọi là tăng tốc độ tuyến tính nếu tốc độ tăng là N khi hệ thống lớn có N lần tài nguyên (CPU, đĩa ...) lớn hơn hệ thống nhỏ. Nếu tốc độ tăng nhỏ hơn N, hệ thống được gọi là tăng tốc độ hạ tuyến tính (sublinear).

Tăng quy mô liên quan đến khả năng xử lý các công việc lớn trong trong cùng một lượng thời gian bằng cách cung cấp thêm tài nguyên. Giả sử, Q là một công việc, Q_N là một công việc N lần lớn hơn Q. Giả sử thời gian thực hiện công việc Q trên một máy M_S là T_S và thời gian thực hiện công việc Q_N trên một máy song song M_L, N lần lớn hơn M_S, là T_L. Tăng quy mô được định nghĩa là $\mathbf{T_S/T_L}$. Hệ song song M_L được gọi là tăng quy mô tuyến tính trên công việc Q nếu $\mathbf{T_S = T_L}$. Nếu $T_L > T_S$, hệ thống được gọi là tăng quy mô hạ tuyến tính. Tăng quy mô là một độ đo (metric) quan trọng hơn trong đo lường hiệu quả của các hệ CSDL song song. Đích của song song trong các hệ CSDL là đảm bảo hệ CSDL có thể tiếp tục thực hiện ở một tốc độ chấp nhận được, ngay cả khi kích cỡ của CSDL và số giao dịch tăng lên. Tăng khả năng của hệ thống bằng cách tăng sự song song cung cấp một con đường thuận tiện hơn cho sự phát triển hơn là thay thế một hệ tập trung bởi một máy nhanh hơn. Một số nhân tố ảnh hưởng xấu đến tính hiệu quả của hoạt động song song và có thể làm giảm cả tăng tốc độ và tăng quy mô là:

- *Chi phí khởi động (Startup Costs):* Có một chi phí khởi động kết hợp với sự khởi động một xử lý. Trong một hoạt động song song gồm hàng ngàn xử lý, thời gian khởi động (Startup time) có thể làm lu mờ thời gian xử lý hiện tại, ảnh hưởng bất lợi tới tăng tốc độ.
- *Sự giao thoa (Interference):* Các xử lý thực hiện trong một hệ song song thường truy nhập đến các tài nguyên chia sẻ, một sự giao thoa của mỗi xử lý mới khi nó cạnh tranh với các xử lý đang tồn tại trên các tài nguyên bị chiếm như bus hệ thống, đĩa chia sẻ, thậm chí cả đồng hồ. Hiện tượng này ảnh hưởng đến cả tăng tốc độ lẫn tăng quy mô.
- *Sự lệch (Skew):* Bằng cách chia một công việc thành các bước song song, ta làm giảm kích cỡ của bước trung bình. Tuy nhiên, thời gian phục vụ cho bước chậm nhất sẽ xác định thời gian phục vụ cho toàn bộ công việc. Thường khó có thể chia một công việc thành các phần cùng kích cỡ, như vậy cách mà kích cỡ được phân phối là bị lệch. Ví dụ: một công việc có kích cỡ 100 được chia thành 10 phần và sự phân chia này bị lệch, có thể có một số phần có kích cỡ nhỏ hơn 10 và một số nhiệm vụ có kích cỡ lớn hơn 10, giả sử trong đó có phần kích cỡ 20, độ tăng tốc nhận được bởi chạy các công việc song song chỉ là 5, thay vì 10.

- **Các mạng hợp nhất (Interconnection Network):** Các hệ thống song song gồm một tập hợp các thành phần (Processors, memory và các đĩa) có thể liên lạc với nhau thông qua một mạng hợp nhất. Các ví dụ về các mạng hợp nhất là:
  - *Bus:* Toàn bộ các thành phần hệ thống có thể gửi và nhận dữ liệu qua một bus liên lạc. Các kiến trúc bus làm việc tốt với một số nhỏ các processor. Tuy nhiên, chúng không có cùng quy mô khi tăng sự song song vì bus chỉ điều khiển liên lạc từ chỉ một thành phần tại một thời điểm.

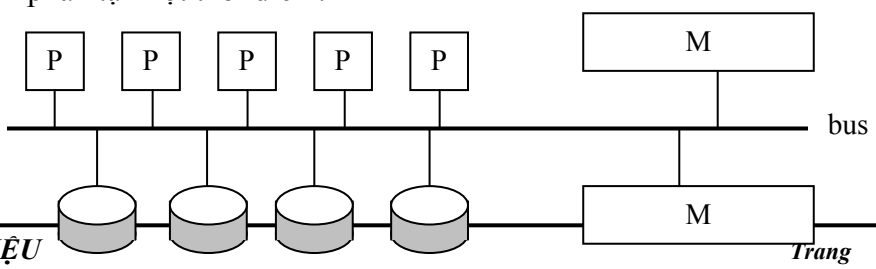
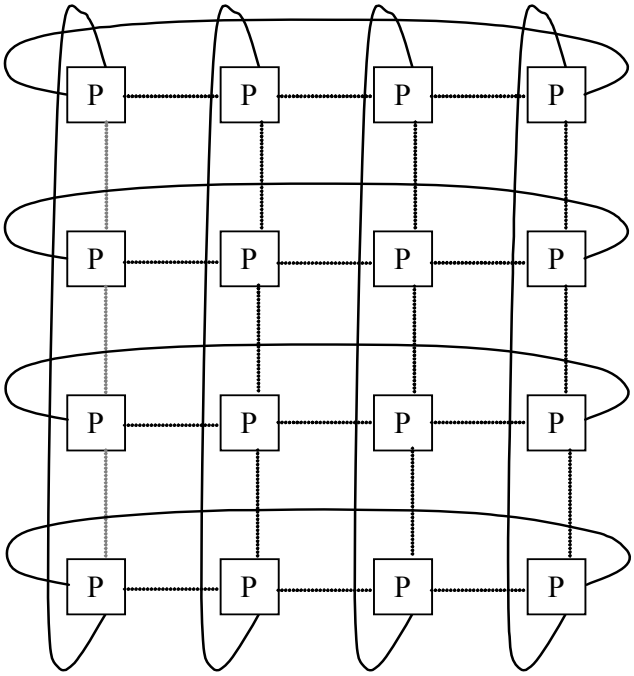


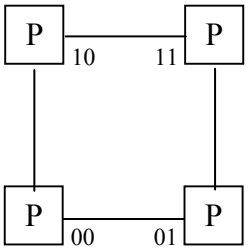
Figure 5

- *Lưới (Mesh):* Các thành phần được sắp xếp như các nút của một lưới, mỗi thành phần được nối với tất cả các thành phần kề với nó trong lưới. Trong một lưới hai chiều mỗi nút được nối với 4 nút kề.

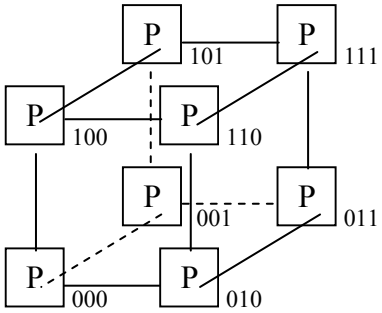


mạng hợp nhất hình lưới

Siêu lập phương một chiều



Siêu lập phương hai chiều



Siêu lập phương ba chiều

Figure 6

- *Siêu lập phương (Hypercube)*: Các thành phần được đánh số theo nhị phân, một thành phần được nối với thành phần khác nếu biểu diễn nhị nhân số của chúng sai khác nhau đúng một bit. Như vậy, mỗi một thành phần trong n thành phần được nối với log(n) thành phần khác. Có thể kiểm nghiệm được rằng trong một sự hợp nhất siêu lập phương một thông điệp từ một thành phần đến một thành phần khác đi quá nhiều nhất log(n) nối kết, còn trong lưới là $\sqrt{n}-1$

- **Các kiến trúc cơ sở dữ liệu song song:** Có một vài mô hình kiến trúc cho các máy song song:

  (Ký hiệu: P = processor, M = Memory, = đĩa)

  - *Bộ nhớ chia sẻ (Shared memory)*: Tất cả các các processor chia sẻ một bộ nhớ chung. Trong mô hình này các processor và các đĩa truy xuất một bộ nhớ chung, thường thông qua một bus hoặc một mạng hợp nhất. Thuận lợi của chia sẻ bộ nhớ là liên lạc giữa các processor là cực kỳ hiệu quả: dữ liệu trong bộ nhớ chia sẻ có thể được truy xuất bởi bất kỳ processor nào mà không phải di chuyển bởi phần mềm. Một processor có thể gửi thông điệp cho một processor khác bằng cách viết vào bộ nhớ chia sẻ, cách liên lạc này nhanh hơn nhiều so với các liên lạc khác. Tuy nhiên, các máy bộ nhớ chia sẻ không thể hỗ trợ nhiều hơn 64 processor vì nếu nhiều hơn, bus hoặc mạng hợp nhất sẽ trở nên dễ bị nghẽn (bottle-neck). Kiến trúc bộ nhớ chia sẻ thường có những cache lớn cho mỗi processor, như vậy việc tham khảo bộ nhớ chia sẻ có thể tránh được mỗi khi có thể.

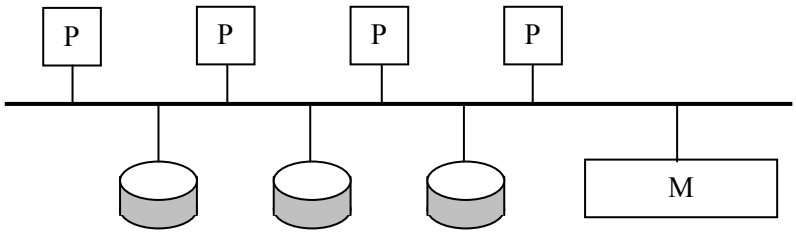


Figure 7

- *Đĩa chia sẻ (Shared disk):* Tất cả các processor chia sẻ đĩa chung. Mô hình này còn được gọi là cụm (cluster). Trong mô hình này tất cả các processor có thể truy xuất trực tiếp đến tất cả các đĩa thông qua một mạng hợp nhất, nhưng mỗi processor có bộ nhớ riêng.

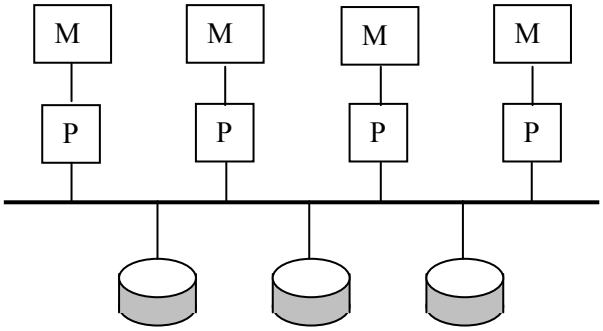


Figure 8

Kiến trúc này có các điểm thuận lợi là: thứ nhất, bus bộ nhớ không bị bottle-neck, thứ hai, cho một phương pháp rẻ để cung cấp một mức độ lượng thứ lỗi--một processor bị hỏng hóc, các processor khác có thể tiếp tục công việc của nó. Ta có thể tạo ra hệ thống con các đĩa tự lượng thứ lỗi bằng cách sử dụng kiến trúc RAID (được trình bày sau này). Vấn đề chính của chia sẻ đĩa là sự hợp nhất các hệ thống con các đĩa trở nên bottle-neck, đặc biệt trong tình huống CSDL truy xuất đĩa nhiều. So sánh với bộ nhớ chia sẻ, chia sẻ đĩa có thể hỗ trợ một số lượng processor lớn hơn, nhưng việc liên lạc giữa các processor chậm hơn.

- *Không chia sẻ (Shared nothing):* Các processor không chia sẻ bộ nhớ chung, cũng không chia sẻ đĩa chung. Trong hệ thống này mỗi nút của máy có một processor, bộ nhớ và một vài đĩa.

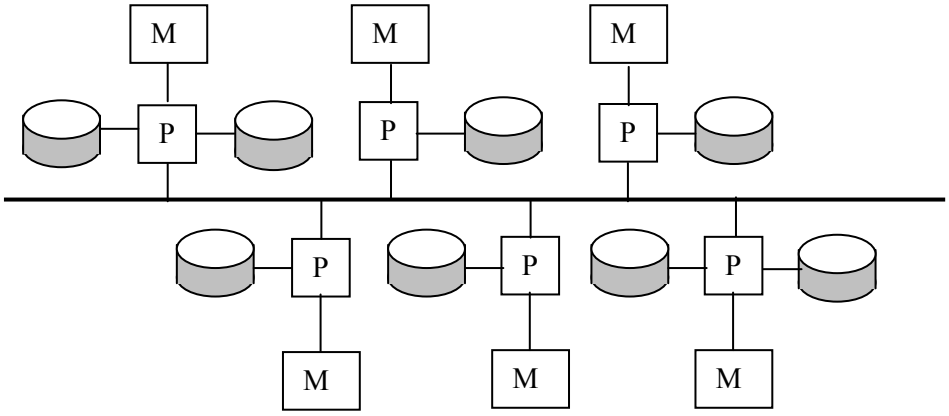


Figure 9

Các processor ở mỗi nút có thể liên lạc với các processor khác qua mạng hợp nhất tốc độ cao. Chức năng của một nút, như server, dữ liệu được chứa trên các đĩa của nó. Mô hình không chia sẻ gì chỉ có vấn đề về việc truy xuất các đĩa không cục bộ và việc truyền các quan hệ kết quả qua mạng. Hơn nữa, đối với các hệ thống không chia sẻ gì, các mạng hợp nhất thường được thiết kế để có thể tăng quy mô, sao cho khả năng truyền của chúng tăng khi các nút mới được thêm vào.

- *Phân cấp (hierarchical):* Mô hình này là một sự lai kiểu của các kiến trúc trước.

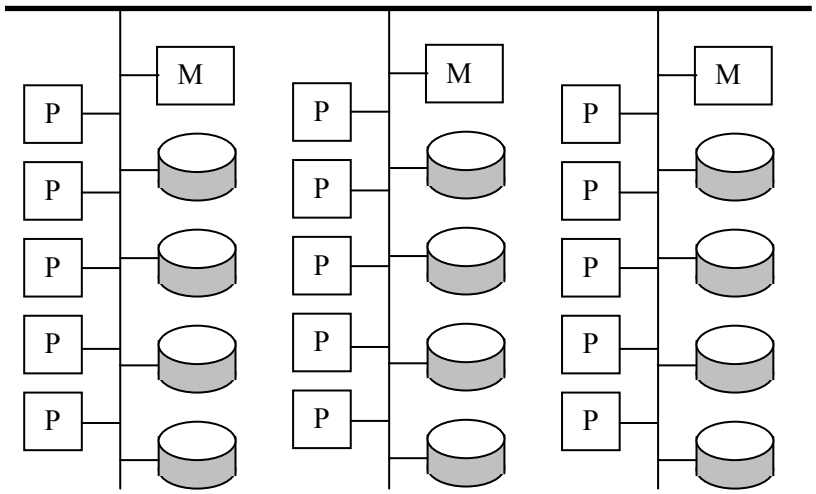


Figure 10

Kiến trúc này tổ hợp các đặc trưng của các kiến trúc chia sẻ bộ nhớ, chia sẻ đĩa và không chia sẻ gì. Ở mức cao nhất, hệ thống bao gồm những nút được nối bởi mạng hợp nhất và không chia sẻ đĩa cũng như bộ nhớ với nút khác. Như vậy, mức cao nhất là kiến trúc không chia sẻ gì. Mỗi nút của hệ thống có thể là hệ thống chia sẻ bộ nhớ với một vài processor. Kế tiếp, mỗi nút có thể là một hệ thống chia sẻ đĩa. Mỗi một hệ thống chia sẻ đĩa lại có thể là một hệ thống chia sẻ bộ nhớ... Như vậy, hệ thống có thể được xây dựng như một sự phân cấp.

CÁC HỆ THỐNG PHÂN TÁN (Distributed Systems):

Trong một hệ thống CSDL phân tán, CSDL được lưu trữ trên một vài máy tính. Các máy tính trong một hệ thống phân tán liên lạc với một máy khác qua nhiều dạng phương tiện liên lạc khác nhau: mạng tốc độ cao, đường điện thoại... Chúng không chia sẻ bộ nhớ cũng như đĩa. Các máy tính trong hệ thống phân tán có thể rất đa dạng về kích cỡ cũng như chức năng: từ các workstation đến các mainframe. Các máy tính trong hệ thống phân tán được tham chiếu bởi một số các tên khác nhau: site , node -- phụ thuộc vào ngữ cảnh mà máy được đề cập. Ta sẽ sử dụng thuật ngữ **site** để nhấn mạnh sự phân tán vật lý của các hệ thống này.

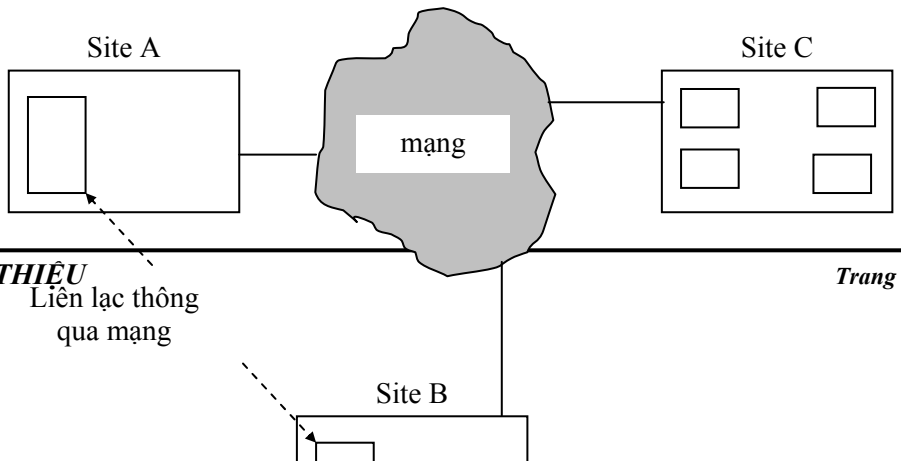


Figure 11

Sự sai khác chính giữa CSDL song song không chia sẻ gì và hệ thống phân tán là CSDL phân tán được tách biệt về mặt địa lý, được quản trị tách biệt và có một sự hợp nhất chậm. Hơn nữa, trong hệ thống phân tán người ta phân biệt giữa các giao dịch cục bộ (local) và toàn thể (global). Giao dịch cục bộ là một giao dịch truy xuất dữ liệu trong một **site** tại đó giao dịch đã được khởi xướng. Giao dịch toàn thể là một giao dịch mà nó hoặc truy xuất dữ liệu trong một site từ một site khác tại đó nó được khởi xướng hoặc truy xuất dữ liệu trong một vài site khác nhau.

BÀI TẬP CHƯƠNG I

I.1 Bốn điểm khác nhau chính giữa một hệ thống xử lý file và một hệ quản trị CSDL là gì ?

I.2 Giải thích sự khác nhau giữa độc lập dữ liệu vật lý và độc lập dữ liệu logic

I.3 Liệt kê năm nhiệm vụ của nhà quản trị CSDL. Đối với mỗi nhiệm vụ, giải tích rõ những vấn đề nảy sinh nếu nhiệm vụ đó không được hoàn thành.

I.4 Năm chức năng chính của nhà quản trị CSDL là gì ?

I.5 Sử dụng một mảng hai chiều kích cỡ n x m như một ví dụ để minh hoạ sự khác nhau:

1. giữa ba mức trừu tượng dữ liệu
2. giữa sơ đồ và thể hiện

I.6 Trong các hệ thống client-server tiêu biểu, máy server thường mạnh hơn máy client rất nhiều: Có bộ xử lý nhanh hơn thậm chí có thể có nhiều bộ xử lý, có bộ nhớ lớn hơn, có dung lượng đĩa lớn hơn. Ta xét một kịch bản trong đó các máy client và các máy server là mạnh như

nhau. Xây dựng một hệ thống client-server theo kịch bản như vậy có ưu nhược điểm gì ? Kịch bản nào phù hợp hơn với kiến trúc server dữ liệu ?

I.7 Giả sử một giao dịch được viết trong C với SQL nhúng, và khoảng 80% thời gian được dùng cho code SQL, 20% còn lại cho code C. Nếu song song được dùng chỉ cho code SQL, Tăng tốc độ có thể đạt tới bao nhiêu ? Giải thích.

I.8 Những nhân tố nào chống lại việc tăng quy mô tuyến tính trong một hệ thống xử lý giao dịch ? Nhân tố nào là quan trọng nhất trong mỗi một kiến trúc sau: bộ nhớ chia sẻ, đĩa chia sẻ, không chia sẻ gì ?

I.9 Xét một mạng dựa trên đường điện thoại quay số tự động, trong đó các site liên lạc theo định kỳ, ví dụ hàng đêm. Các mạng như vậy thường được cấu hình với một site server và nhiều site client. Các site client chỉ nói với server và trao đổi dữ liệu với client khác bởi lưu dữ liệu tại server và lấy dữ liệu được lưu trên server bởi client khác. Ưu, nhược điểm của kiến trúc như vậy là gì ?