Database Machines

A *database machine* takes some pressure off the computer on which the application programs run, so that the total capacity of the machine’s configuration is increased. *Client – Server Architecture.*

May consist of special purpose mini, micro computer, or other.

Increasing a db machine’s capacity is often cheaper than building a large application computer

1. **Architecture of a db Machine**
Distribution of software functions between a DB machine and an application machine

1. If a program needs to read a record in the db, a db subroutine is called.

2. The application program itself must be independent of the db machine hardware and software, therefore the db subroutine has to know whether a db subroutine is used or not. If a db machine is used, the db statement is passed on to the communication system.

3. The communication system contains the basic components of a general Transaction Management System (TMS). Since its task is to manage the physical connection between the two computers. (The communication system must have a communication protocol and a system for detecting and correcting errors, managing the buffers, etc.) By means of this software the db statements of the application program can be transferred to the db machine.

4. When a db machine receives the db statement a task for the statement is created; i.e. a program that analyzes the db statement is started.

5, 6. Normally the result of the above analysis is that the DBMS is called on once or several times, corresponding to the records the db statement needs to read.

7. After the data belonging to the db statement is located, it is passed on the communication system of the db machine.

8. User view records are transferred from the db machine to the communication system of the application machine.

9. User view records are transferred to the buffer area of the application program, after which control is passed back to the application program.
2. Multiple Machines with a Control Machine

Another architecture is to build a db machine from a number of different db machines. So the db machine consists of several sub-machines together with a control machine.

There are two different types:

- **Distribution by Data**: Each individual db machine has to control a specific section of the db.
- **Distribution by Function**: Each individual DB machine is specialized to perform a specific task (i.e. reading, writing, updating)

**Multiple Data Base Machine Architecture**
OTHER CONCURRENCY CONTROL PROTOCOLS

1. **Thomas’ Write Rule.**
   This is a modification of a time stamp protocol in which obsolete or outdated (i.e. if \( ts(T) < write-ts(I) \) – see timestamps for DI’s) write operations may be ignored under certain circumstances. The protocol rules for read are unchanged, but for write they are different.

2. **Validation Techniques.**
   Here, because the majority of Transactions are read-only Transactions, then the rate of conflict among these Transactions may be very low.

3. **Multiversion Concurrency Control.**
   This protocol assumes that each write(I) operation creates a new version of T. When a read(I) operation is issued, the system selects one of the versions of I to be read. A read operation always succeeds, while a write operation may result in Rolling Back the Transaction.

4. **The Tree Protocol.**
   This is similar to the wait-for graphs. Several data items I are grouped together and treated as one aggregate I resulting in hierarchy (tree).

5. **The Phantom Phenomenon.**
   (Please see this protocol in the Concurrency Problems.)
   This protocol fixes that problem.