DDS Assignment 3 - An interpretation of DBMS Musing by Daniel Abadi

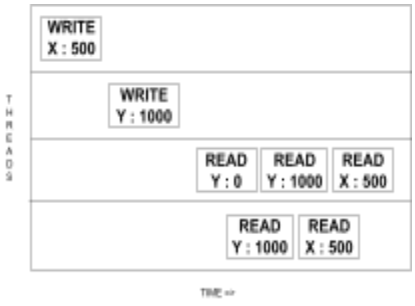
By: Akshay Malhotra

Introduction: In this article, we learn about consistency levels associated with distributed database systems. Users get to choose between different consistency levels (indirectly linked to performance) when setting up their databases.

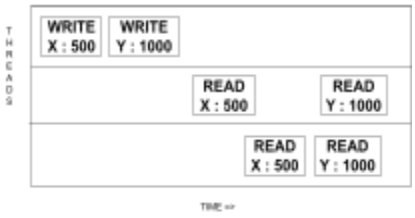
Understanding Consistency Levels: Consistency in databases is context-dependent. There are 2 possible contexts: (1) Applications complying with ACID properties (2) Applications complying with CAP theorem. Consistency in ACID properties refers to applying referential integrity constraints, primary and foreign keys constraints and application constraints whereas consistency in CAP theorem refers to retrieving (read) the most recent value of a write operation irrespective of where the read or write is performed. The concept of 'consistency levels' is never associated with applications that comply with ACID properties. This is because achieving consistency in such applications is mostly the responsibility of the application developer. Instead, 'consistency levels' are associated with applications that comply with CAP theorem. An application with 'strict consistency' will retrieve the value of the most recent write (irrespective of location or server). This comes with a trade-off with performance and availability. Thereby, database systems should be designed carefully with requirements of performance in mind.

Types of Consistency Levels

1. Sequential Consistency
    - All writes can be globally ordered (irrespective of thread id or data object).
    - All threads in context agree to global ordering.
    - No rules with respect to global ordering - can be different from real-time ordering.



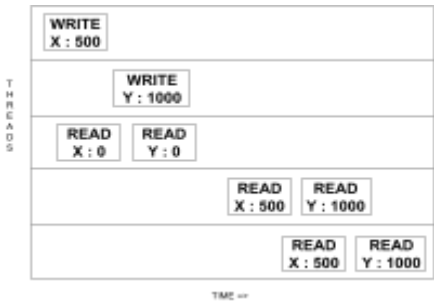
2. Strict Consistency (extension of Sequential Consistency)
    - All threads in context agree to global ordering.
    - Every thread agrees to the global current time.
    - Strict rule with respect to global ordering - should be same as real-time ordering.
    - Retrieval (read) of data object always returns most recent write value regardless of server.



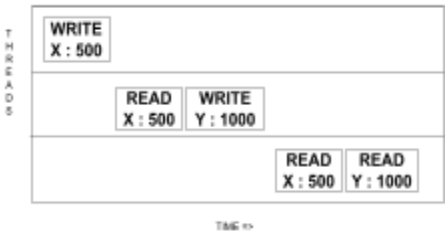
DDS Assignment 3 - An interpretation of DBMS Musing by Daniel Abadi

By: Akshay Malhotra

3. Atomic Consistency / Linearizability (extension of Sequential Consistency)
    - All threads in context agree to global ordering.
    - Every thread agrees to the global current time.
    - Acknowledges existence of request service time - the time between submission of the write request and acknowledgment message.
    - Strict rule with respect to global ordering - should be the same as real-time ordering except for overlapping operations.
    - Retrieval (read) of data object always returns most recent write value regardless of server.



4. Causal Consistency
    - All threads in context do not have to agree to global ordering.
    - Every thread independently agrees to the local ordering of operations enforcing a casual constraint.



5. Eventual Consistency
    - The most recent value is retrieved by all threads if no writes are performed for extended periods of time and changes are eventually reflected on all servers.

An important feature of consistency levels and transactions is that a database moves through a sequence of globally defined states. This helps us achieve abstraction whereby the end-user isn't informed about the complications of replication taking place underneath. In the case of a highly consistent database, there is only one view that gives the illusion of a centralized database system. For lower consistent databases, there can be multiple views of the database as it transitions through states. This increases the complexity of management and development for an application developer.

A transaction in a database can have multiple reads and writes. In order to map consistency levels to transactions, we need to identify each operation in the database with a transaction id and draw a box around the transaction. AID properties of the ACID properties apply to transactions, ie, the transactions have to be atomic, isolated and durable.