Two Phase locking guarantees serializability

Proof is by contradiction.

Suppose the 2PL schedule or history is not serializable. It means the conflict graph has a cycle.

That means a transaction has released a lock (to cause an outgoing edge) and then asked for a lock (to cause an incoming edge)

See theorem 11.2 in

https://www.cs.purdue.edu/homes/bb/cs448s24/syllabus/1h.pdf

Optimistic CC is based on detecting a cycle after a transaction reads, writes, and try to commit.

If a cycle occurs with committed transactions, we abort this transaction or else commit it. Any future ongoing transaction will have to check for a cycle with all committed transactions that were running in parallel with the transaction that wants to commit.

The idea is not to block small transactions from committing and make them wait (due to locks held by a long transaction).

Optimistic performs well when there is a mix of small and large transactions.

Performance measures for a concurrency control algorithm.

a) Degree of concurrency (how many transactions can run concurrently without getting blocked or aborted)
b) How many deadlocks occur?
c) How many transactions are aborted in optimistic?
d) Overhead of locking
e) Overhead of detecting cycle in optimistic
f) Average Response time of transaction execution
g) Throughput provided (transactions committed per second)
   Etc

Ideas for recovery

Logs, undo/redo logs, write-ahead logging.
Rollbacks
Deferred updates
Backup of data (Multiple old versions)
Differential updates (updates go to another file and not on main database) (In place versus shadow updates
Checkpoints of transactions, snapshots of databases.
Identifying data updates that have caused inconsistencies or integrity violations.
All pages updated by a transaction are immediately written to disk before the transaction.
Commits
Logs and backup data stored at physically remote sites.
Two phase commitment (first phase to prepare to commit and then in second phase commit to permanent storage)

Performance Measures of Recovery control

(a) Time needed to do backups, create logs, take checkpoints.
(b) Time needed for recovery bring data back to a consistent state)
(c) How much data is corrupted?
(d) Overhead on response time, throughput of transaction processing
(e) Can transaction processing continue while backups and maintenance are being done?
(f) How many and what type of failures can the system tolerate and continue processing without halting the system?
(g) Etc.

Ideas for Enforcing Privacy of data.

Encryption/Decryption of data
Increasing anonymity (disclosing a few data items where a very large number of items exist)
Data Access control policy for individual user (Role based control)
Hiding data
Giving statistical values such as average, median, mode etc. rather than individual values
Differential privacy (not answering multiple queries to multiple or same user on same data in a short period of time)

Performance Measures of data privacy

a) Amount of sensitive Data disclosed. (Number of data items and their values)
b) Are exact values disclosed or approximate or statistical values disclosed?
c) Is Data disclosed from a small database or large database?
d) Has data been disclosed to people with wrong security clearances (such as top secret, secret, confidential or unclassified in military)
e) Is privacy violated and data given to one person or all persons (publicly leaked)
f) What mechanisms can be deployed to protect data after some leak?
g) Etc.