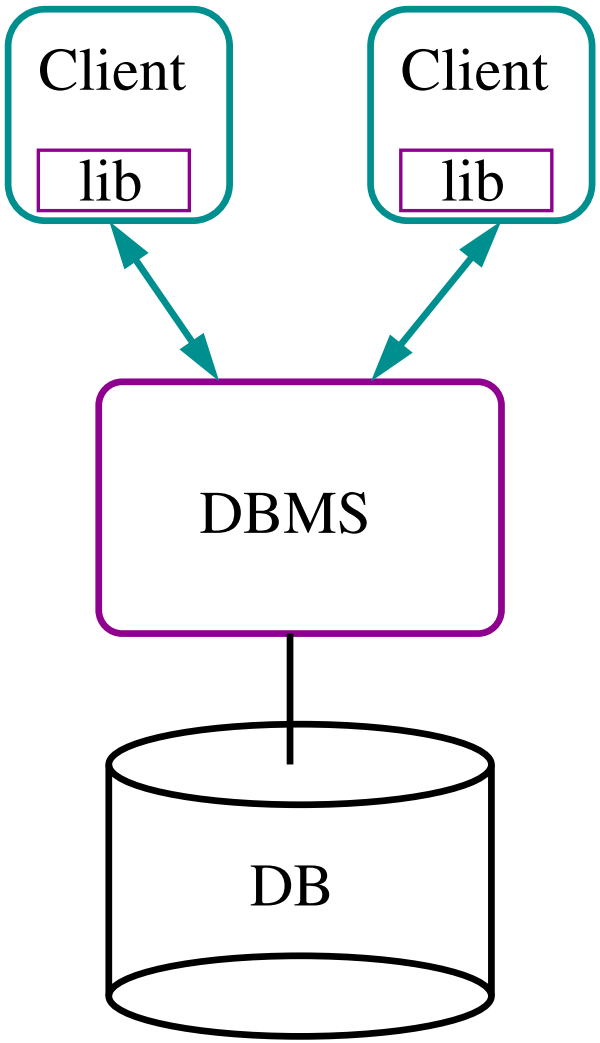
# DBMS Clients and Server




- client API implemented by DBMS client library
- client relational APIs: ODBC, JDBC, proprietary, . . .
- client session:
  - connect
  - issue one or more commands, e.g., queries
  - disconnect
- some DBMS additionally implement stored procedures, server-side programs that issue commands, and that are invoked by the client

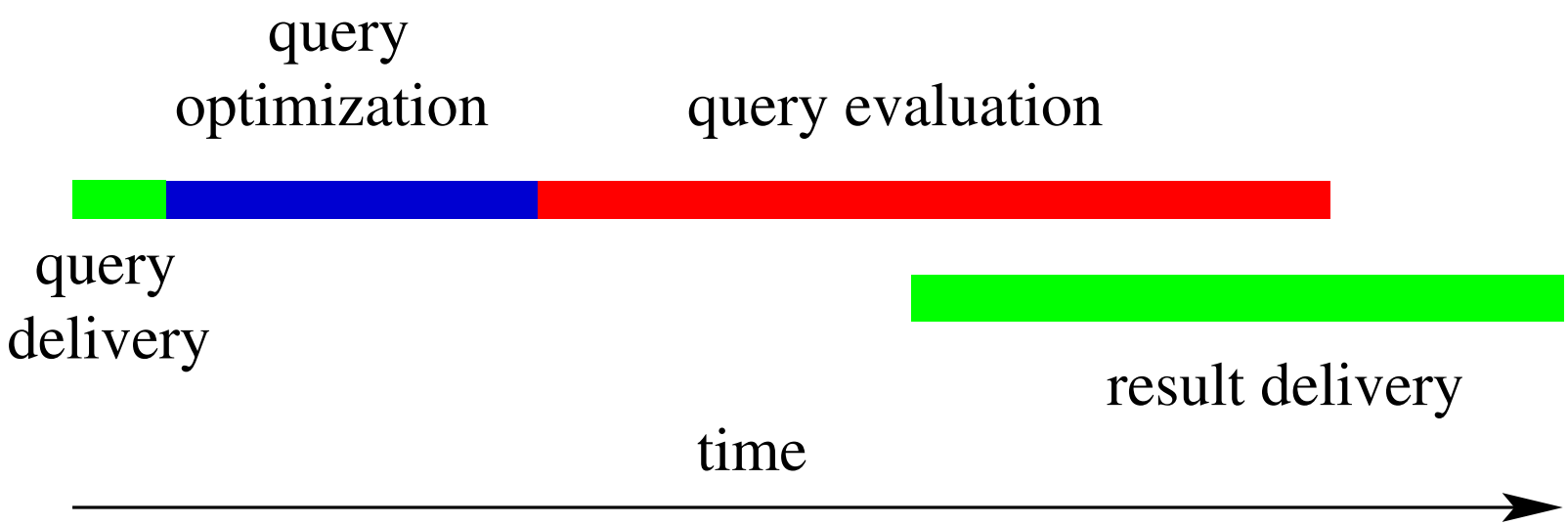
# Simplified Workflow

1. Client establishes a connection.
2. Client issues a request (e.g., a query) over the connection.
3. DBMS process manager assigns a worker process/thread to handle the request.
4. Worker chooses an optimized execution plan for the query and executes the plan.
5. As the execution plan runs, it accesses stored data from the database and generates results.
6. Results are passed back to the client over the connection.

# Query Processing: Big Picture

| Query Processing and Evaluation |
| --- |
| Relational Operators |
| Access Methods/Indexing |
| Buffer Management |
| Storage Management |

# Performance

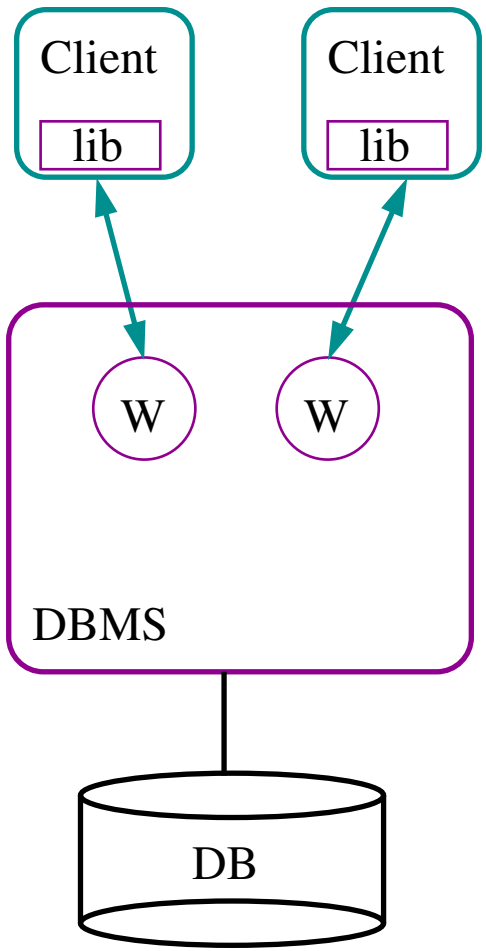


# DBMS Server Process Architecture

- handling client requests
  - worker thread (or process) per connection (e.g., PostgreSQL)
  - worker thread (or process) per request: thread pool, dispatcher
- DBMS may use additional threads for specialized tasks, e.g.,
  - prefetching data into memory
  - page cleaning
- some DBMS support parallel query execution, which allows multiple threads to work in parallel on a single query

# Worker Threads/Processes

## Thread Per Client




## Thread Pool