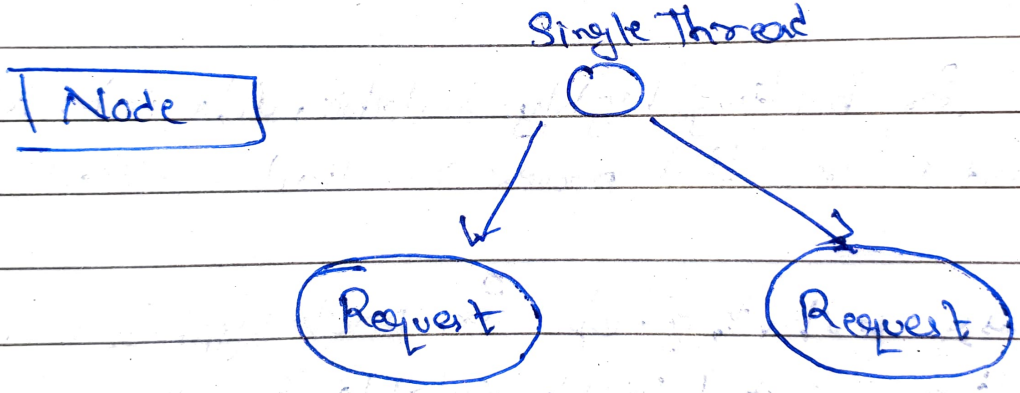
* How Node works:

⇒ Non-blocking / Asynchronous architecture:

A single thread is used to handle multiple requests.

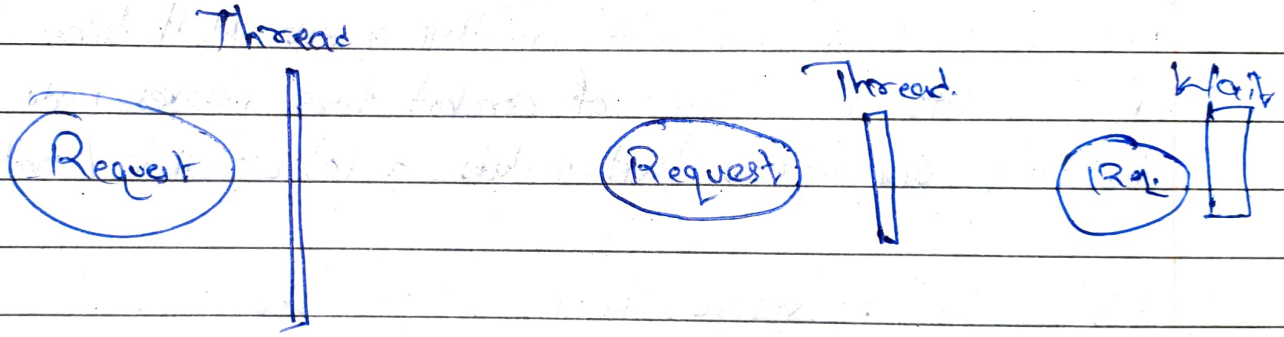


→ Blocking / Synchronous Architecture:

When we request on the server a thread is allocated to handle that request. (As part of handling that request its likely that we're going to query a database (sometimes it may take little while until the result is ready).

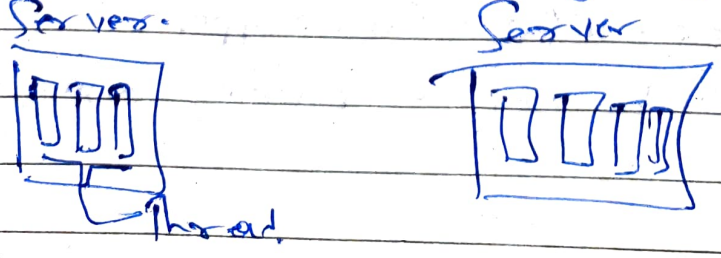
When the DB is executing the query that thread is sitting there waiting it can't be used to server another client.

So we need a new thread to serve another client.



If we have large no. of client & all thread are been used then new clients have to wait until free threads are available / if we don't want to wait we add more hardwa

These type of architecture, we're not utilizing our resources efficiently. the problem with blocking / synchronous archit



→ Node applications are asynchronous by default.

In node we've single thread to handle all request.

If we need to create a DB / thread doesn't have to wait for the DB to return the data. While the DB is executing our query that thread will be used to servers another client.
When the DB prepares the result it puts a message in Event Queue.

Node is continously monitoring this event in background. when it finds an event in this queue it'll take it out & process it, this type of architecture makes node ideal for building applications that include a lot of disk / network access.

We can servers more clients without the need to throw in more hardware.

Node is ideal for I/O intensive apps.

Do not use Node for CPU-intensive apps. (video encoding, etc.)

Since it's single thread, another client has to wait so it should not be used for CPU intensive apps.
It should only be used for building data intensive & real-time applications.

→ V8 is a j.s. engine i.e. it parses & executes Js code.