Extracted from VLDB'07 (Stonebraker et al.; the End of an Architectural Era. It's time for a complete rewrite)

2nd BOTTLENECK: JDBC Connectors

Getting transactions into and out of the system is likely to be the next significant bottleneck. The overhead of JDBC/ODBC style interfaces will be onerous, and something more efficient should be used. In particular, we advocate running application logic – in the form of stored procedures – "in process" inside the database system, rather than the inter-process overheads implied by the traditional database client / server model.
This bottleneck is related to what is called as **"the impedance mismatch"**.
The object-relational impedance mismatch is a set of conceptual and technical difficulties that are often encountered when a RDBMS is being used by a program written in an object-oriented programming language. Particularly, it refers to the time lost mapping objects or class definitions to database tables.

3rd BOTTLENECK: Latching

The latching associated with multi-threaded data structures is likely to be onerous. In front of short runtime of transactions, moving to a single threaded execution model will eliminate this overhead at little loss in performance.

Latch definition: A simple, low-level serialization mechanism to protect shared data structures in the system global area.