The paper presents a new distributed database called SALT, for improving the scalability and performance of the ACID transactions, using the BASE approach.The design of the SALT database follows the Pareto principle. When looking at performance requirements of the ACID transactions, we find that there are only a few which require high performance. To handle these transactions, the SALT database introduces a new workflow, using BASE approach. These BASE transactions provide the required properties of durability and atomicity. Also, new SALT isolation provides control at the level of isolation, for other ACID or BASE transactions. The primary reason for the tradeoffs between the performance and ease of programming is providing the same level of scaling for the atomicity and isolation. Thus, this BASE approach loosens the coupling between the isolation and the atomicity.

To find the different transactions that are loaded on the system, the paper increases the load on the system, and finds which of the transactions cause anomaly latencies. BASE transactions are described in alkaline subtransactions. These alkaline sub transactions are executed atomically. Salt isolation is offered by the BASE transactions, which provides multiple isolation granularities, thereby making states of BASE transactions accessible in limit. However, SALT guarantees the same level of the isolations for ACID, as in a pure ACID database.

The paper compares the new SALT system with existing ACID transactions to check if the BASE transactions can provide any performance improvements. SALT solves one of the issues of the interleaving transactions, which occurs when a transaction is divided into multiple sub transactions. To manage the concurrency, the SALT system uses locks mechanism as follows: ACID for ACID transactions, Alkaline for sub transactions and ACID, Saline for BASE transactions. Benchmarking is done using the TPC-C database benchmark, on the MySQL Cluster, and using the popular fusion ticket application. The evaluations of the SALT system, tries to answer two main questions: the performance gain difference with respect to the ACID systems, and how much is the SALT system affected by different work-loads.

Strong Points:
1). The committed state of each of the alkaline subtransactions can be observed by the other BASE or subalkaline transactions, thereby achieving higher performance and availability than in the ACID database. Thus, SALT provides around 80% more throughput performance when used in systems with all BASE transactions.
2). The paper provides legible proofs and theorems to prove the results of the SALT system.
3). Providing SALT isolation, guarantees the concurrency for BASE transactions, and isolation for the ACID transactions.

Weak Points:
1). SALT requires finding out transactions which are bottlenecks in the performance, this may lead to more overhead for developers.

2). Developing solutions for the alkaline, BASE and ACID transaction management is another overhead for developers, as the developer has to keep the database in a consistent state after each of the transactions.
3). Also, providing 3 different lock mechanisms is tedious for the developers to handle.

Question:
1). The paper mentions the concept of interleaving, when trying to divide any transactions. What does interleaving mean?
2). What can be the implications of using such systems in a partition database system?