

# Bibliography

- [1] Michael Stonebraker, Ugur Çetintemel: "One Size Fits All": An Idea Whose Time Has Come and Gone (Abstract). ICDE 2005: 2-11

In this 2005 paper, authors argue RDBMS is not suitable for data ware house and streaming applications. Traditional RDBMS is optimized for writes whereas data ware house applications should be read-optimized because they periodically pulls huge amount of data from different sources and complex analytic queries are then run offline. The database schema for warehouses or OLAP (online analytic processing app) looks like a star: a central facts table containing one row of info per event and many dimension tables containing details about the various field of facts; this star schema is not typical in OLTP (online transaction processing) using RDBMS. OLAP uses bit-map indexes and OLTP uses B-tree indexes. In a streaming application there are three main parts. (1) application server (2) data bus (3) storage - preferably all residing inside a single address space (single-process, multi-threaded). These apps can tolerate occasional lost updates so do not require ACID guarantees. In short, the authors foresaw the rise of NoSQL aka not only SQL.

- [2] Heimerdinger. Walter, and Weinstock. Charles, "A Conceptual Framework for System Fault Tolerance," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-92-TR-033, 1992.

A system is composed of components and it exists within an environment; it provides feedbacks to its operators and service to its users. A dependable system is available, reliable, safe (absence of catastrophies), and secure. Fault tolerance is a means to achieve dependability. In order to talk about fault tolerance, we need to agree on a fault vocabulary. A system has failed if the service it delivers to its users, does not comply with the system specification for a specified amount of time. The system specification defines the boundary of concerns. Every system has an explicit and an implicit specification; the latter comes from the expectation user has from her experience with other similar systems. Faults cause failure. When a fault crosses the system boundary—therefore becomes user-visible—it is a failure. The fault floor is where region of concern ends. Fault trajectory is the sequence of faults that leads to a failure. For easier manageability, faults can be classified according to locality, effect on time or data, cause, duration, and effect on system state. To be fault tolerant a system must detect, diagnose, contain, mask, compensate, and recover from faults. To detect faults we need redundant information—the information that would not be needed if faults were absent. Core of fault tolerance is redundancy management, for example failover the primary replica to a back-up replica.