



Databases in 21st Century: NimbusDB and the Cloud.

Jim Starkey, CTO and Founder, NimbusDB Inc.

An old adage advises getting your head out of the clouds and your feet on the ground. The exploding numbers of mobile devices and applications and the exponential growth of social networking make this piece of wisdom a candidate for Adage 2.0: "Keep your head in the clouds, but stretch until your feet are on solid ground".

Three aspects of cloud computing challenge traditional database technology. The first is scalability. No single computer system can support the loads imposed by popular worldwide applications. Worse, loads cannot be predicted accurately and configuring for the greatest possible load is economically catastrophic. The second challenge is multi-tenancy. Amazon, Google, and Rackspace demonstrate that a well-managed multi-tenant cloud delivers excellent performance and high resource utilization. Traditional database systems are single tenant; they support a single database per system or per cluster. The third challenge is administration. The cloud environments require strict separation between physical provisioning, which is the province of the cloud vendor, and database administration, the responsibility of the database owner or administrator.

There are many workarounds for each of these challenges. Some offer scalability, without multi-tenancy. Amazon offers a multi-tenant cloud of MySQL databases, restricted to a single machine instance. NoSQL systems offer scalability and multi-tenancy, without ACID transactions, high-level semantics, or declarative consistency.

Traditional database architectures have reached their limits. If we, the database community, are to meet the challenges of computing in the clouds, we need new ideas, architectures, and implementations.



NimbusDB has the features needed for cloud databases:

- **ACID relational database designed from the ground up to stretch into the clouds.**
- **Storage in shared, replicating, demand loaded, distributed objects rather than disk pages.**
- **Concurrency management without central authority, a lock manager, or a transaction scheduler.**
- **Elasticity: when load increases, additional nodes can be added to a running system without interrupting on-going work.**
- **Multi-tenancy: arbitrary numbers of database instances to share processors and storage systems.**
- **Separation between database provisioning and data administration.**
How does NimbusDB work? What makes it different from existing solutions, SQL and noSQL? This presentation describes the reanalysis of relational database technologies that lead to its revolutionary design.

***PLEASE NOTE: The above is an abstract of a talk that Jim Starkey is scheduled to give on Friday, January 28TH at the 2011 New England Database Summit @ M.I.T**