Daniel Abadi
Yale University
dna@cs.yale.edu

I'm interested in building database systems in the cloud. This doesn't really fall squarely in any of the 11 categories listed, but if I have to choose two, they would be:

(1) Cloud Architectures and Systems
(2) Data Portability, Consistency, and Management

My current research in the cloud falls mainly around two publications:

(1) **Data Management in the Cloud: Limitations and Opportunities**
Daniel J. Abadi. In IEEE Data Engineering Bulletin, 32(1), 2009.

(2) **HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads**
Azza Abouzied, Kamil Bajda-Pawlikowski, Daniel J. Abadi, Avi Silberschatz, and Alexander Rasin. In Proceedings of VLDB, 2009.

In these works, we identify key characteristics of database systems built for the cloud, such as horizontal scalability, fault tolerance, and ability to handle wild fluctuations in I/O performance, and propose how database technology can be designed with these goals in mind. The HadoopDB paper mainly focuses on analytical database applications (such as data warehousing and decision support systems). Future work (see the abstract on the next page) revolves around traditional transactional applications.

(Note: if there is not room for me in this workshop, I will not be insulted if I am not invited).

I would like to pursue the problem of horizontally scalable ACID-compliant, transactional database systems. Currently, every database system designed for the cloud is either a sharded version of a single-node database system or is a scalable key-value store. However, there is still demand on behalf of developers for a more traditional relational database solution that can be deployed across an arbitrary number of virtual machines in the cloud. It is much easier for developers if the database can guarantee ACID fully --- otherwise the application layer has to work around these deficiencies which can add significant complexity.

Building an ACID compliant database system for the cloud encompasses several challenges. First, the database system needs to seamlessly take advantage of the elastic scalability that the cloud can offer, which necessitates the ability to partition data across virtual servers (as each server gets more loaded, the data needs to be partitioned at a finer granularity and redistributed across additional virtual servers; similarly, when the servers are less loaded, partitions need to be combined as the system scales down automatically). This necessitates research into fast protocols for atomically committing transactions that span multiple partitions, and, in particular, ensuring that the system remains efficiently utilized while such a commit protocol is ongoing. Furthermore, the system must be able to handle wild fluctuations in virtual server performance, especially for disk I/O which is known to be highly variable in cloud environments. Finally, the system must be highly available, as the cloud is more failure prone than traditional environments, and so the system must make sure the replicas are kept consistent and up-to-date, so that they can be relied upon in the case of a failure. A system that meets all of these requirements will have a significantly different architecture and different concurrency control protocols relative to currently available cloud database systems.