Data replication has always been an important topic. No matter in what context, local access to the data is faster than accessing a remote server. The proliferation of networks has only made this basic principle more important when designing large scale, distributed information systems. Data replication has nevertheless taken many different forms depending on the type of application: caching, lazy replication, eager replication, etc. As the contributions in this special issue demonstrate, the need to provide faster access to dynamic content is blurring these differences in implementation. On the one hand, database caching is becoming the basis for replicating data in remote locations at the edge of the network to provide fast access to dynamic content. On the other hand, data replication techniques are being used to extend the reach of databases beyond the local are network to make them more flexible and applicable in a wider range of contexts. Given the explosive growth of dynamic content and the need for novel solutions in service oriented architectures and commodity computing, data replication will certainly remain an important research topic for years to come.

This issue contains a collection of five papers that address the problem of data replication and caching from a variety of points of view. The first paper, by Theo Härder and Andreas Bümann of the University of Kaiserslautern, addresses the basic issue of populating a cache with the data necessary to answer queries locally. The paper uses the notions of *value completeness*, *value range completeness*, *domain completeness*, and *predicate completeness* to define several fundamental aspects of the database caching problem. The second paper, by Christoph Bornhövd, Mehmet Altinel, C. Mohan, Hamid Pirahesh, and Berthold Reinwald, describes *DBCache*, an extension to DB2 that supports database caching at both the edge of content delivery networks and at the middle-tier of Web site infrastructures. The novelty behind DBCache is that it blurs the notion of data replication and data caching thereby allowing to extend the concept of database caching well beyond conventional applications. The third paper, by Emmanuel Cecchet, presents *C-JDBC*, an open source system that implements the Redundant Array of Inexpensive Databases (RAIDb) concept at the database level. C-JDBC primarily targets J2EE application servers requiring database clusters build with commodity hardware and is therefore an important complement to the idea presented in the first two papers. The fourth paper, by Christian Plattner and Gustavo Alonso, presents *Ganymed*, a flexible data replication solution that exploits snapshot isolation to provide scalability without loss in consistency. Ganymed can be used to implement a database cluster –like C-JDBC– or for caching on the edges of a network –like DBache. The fifth paper, by Per- Åke Larson, Jonathan Goldstein, Hongfei Guo, and Jingren Zhou, describes *MTCache* a mid-tier caching solution for SQL Server. The goal of MTCache is to transparently off-load part of the query workload from a backend server to front-end servers, thereby improving system throughput and scalability but without requiring application changes.