Chapter 25
Data Replication in P2P Systems

Vassilios V. Dimakopoulos
University of Ioannina, Greece

Spiridoula Margariti
University of Ioannina, Greece

Mirto Ntetsika
University of Ioannina, Greece

Evaggelia Pitoura
University of Ioannina, Greece

ABSTRACT
Maintaining multiple copies of data items is a commonly used mechanism for improving the performance and fault-tolerance of any distributed system. By placing copies of data items closer to their requesters, the response time of queries can be improved. An additional reason for replication is load balancing. For instance, by allocating many copies to popular data items, the query load can be evenly distributed among the servers that hold these copies. Similarly, by eliminating hotspots, replication can lead to a better distribution of the communication load over the network links. Besides performance-related reasons, replication improves system availability, since the larger the number of copies of an item, the more site failures can be tolerated. In this chapter we survey replication methods applicable to p2p systems. Although there exist some general techniques, methodologies are distinguished according to the overlay organization (structured and unstructured) they are aimed at. After replicas are created and distributed, a major issue is their maintenance. We present strategies that have been proposed for keeping replicas up to date so as to achieve a desired level of consistency.

INTRODUCTION
In a peer-to-peer (p2p) system, a large number of nodes share data with each other. The participation of nodes is highly dynamic; nodes may enter and leave the system at will. Since, it is not possible to maintain links with all nodes, for performance as well as for privacy and anonymity reasons, each node maintains links with a selected subset of other nodes, thus forming an overlay network. A message
between any two nodes in the p2p system is routed through this overlay network which is built on top of the physical one. Thus, two nodes that are neighbors in the overlay network may be many links apart in the physical network.

The overlay network is built to facilitate the operation of a p2p system. In data sharing p2p systems, a basic functionality is discovering the data of interest. A look-up query for data items may be posed at any node in the overlay. The query is then routed through the overlay to efficiently discover the nodes that hold the requested data items. Such query routing must be achieved by contacting as “small” a number of nodes in the overlay as possible and by maintaining as “little” state information at each overlay node as possible.

There are two basic types of overlays: structured and unstructured ones. To assist lookup, structured overlays map (keys of) data items to nodes. In structured overlays, the mapping is usually done by hashing the key space of the data items to the id space of nodes. Thus, each node in the overlay maintains a partition of the data space. In structured overlays, lookup reduces to locating the node in the overlay that is responsible for the corresponding data partition. In unstructured overlays on the other hand, there is no correlation between nodes and data items.

There are a number of issues regarding the design of a structured p2p system. One design dimension refers to the geometry of the overlay, that is, its structural characteristics. Structured overlays usually follow a regulated topology, such as a ring, tree or grid. Then, upon entering the p2p system, each node takes a specific position in the overlay network. Another design choice is how data are mapped to nodes. The mapping must be fair so that nodes receive similar loads even when the data sets or the operations are skewed. All designs aim at supporting efficiently the basic operations of the overlay, that is, its construction, its incremental maintenance when nodes enter or leave the system, and its search. Efficiency must be achieved even in the case of high churn, where maintaining the overlay structure incurs a high cost. Most structured overlays guarantee lookup operations that are logarithmic in the number of nodes. Finally, overlays differ with respect to the range of different types of queries that they support.

In unstructured overlays, the topology is not a rigid one. Unstructured overlays are formed by the nodes as they join the system by either selecting randomly a node from a known list of participating peers or by following some loose rules regarding this selection. The resulting topology may have certain properties, however there is no assumption regarding the way the data space is mapped to the address space of the nodes in the overlay. To locate data of interest, a node queries its neighbors in the overlay, which in turn query their neighbors, and so on, until the query hits on a node holding the item. However, this procedure provides no guarantees on the complexity of search.

The topology of an unstructured overlay is built up over time in a decentralized manner as peers join and leave the system. In many existing systems, upon joining the network, a peer selects to connect to another peer essentially at random. In these systems, topologies often tend towards a power-law degree distribution, where some long-lived peers have many connections, while most other peers have a few connections.

To improve performance of lookup, caching or replication of either data or search paths (or both) is possible. Besides improving search, replication may also assist in providing load balancing. Further, replication improves fault tolerance and the durability of data items.

Replication increases the number of copies for each shared piece of data in the system. By doing so, the probability that some or all the data is temporarily or permanently lost significantly decreases, thus the dependability of the system in terms of reliability and availability is increased. Additionally, by having more copies for popular data items, the load for routing and answering
Related Content

Adjusting Thread Parallelism Dynamically to Accelerate Dynamic Programming with Irregular Workload Distribution on GPGPUs
www.igi-global.com/article/adjusting-thread-parallelism-dynamically-to-accelerate-dynamic-programming-with-irregular-workload-distribution-on-gpgpus/114710?camid=4v1a

A General Framework for the Modeling and Simulation of Grid and P2P Systems
www.igi-global.com/chapter/general-framework-modeling-simulation-grid/40822?camid=4v1a

City Data Fusion: Sensor Data Fusion in the Internet of Things
www.igi-global.com/article/city-data-fusion/143901?camid=4v1a

Scheduling Large-Scale DNA Sequencing Applications
www.igi-global.com/chapter/scheduling-large-scale-dna-sequencing/36436?camid=4v1a