Effective Dynamic Replication in Wide-Area Network Environments: A Perspective

Anirban Mondal Masaru Kitsuregawa
Institute of Industrial Science,
University of Tokyo,
JAPAN.
{anirban, kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract

The unprecedented growth of data at geographically distributed locations coupled with tremendous improvement in networking capabilities over the last decade strongly motivate the need for efficient data management in wide-area network (WAN) environments such as Peer-to-Peer (P2P) networks and GRIDs. In particular, data availability and performance demands on WAN applications are now greater than ever before. While replication has been traditionally used for maximizing both data availability and performance, this paper contends that replication schemes for traditional distributed environments (e.g., clusters) do not adequately address the requirements of WAN environments. Notably, issues such as node heterogeneity (in terms of processing capacity and available disk space for storing replicas), significant variations in bandwidth, lack of centralized control, lack of global knowledge, distributive ownership and scalability make replication in WAN environments significantly more challenging than in the case of traditional domains. Interestingly, these are fundamental issues which arise for replication in different types of WAN environments. This paper specifically focusses on replication in two representative WAN environments, namely P2P systems and GRIDs, and discusses open research issues concerning replication in these two environments as well as our perspectives on these issues.

1. Introduction

Data has been growing in an unprecedented manner at geographically distributed locations. The emergence of large and powerful computer networks, which have the capability to connect hundreds of thousands of computers worldwide, has created a world of opportunities for global-scale sharing of data. Consequently, data sharing over wide-area networks (WANs), such as Peer-to-Peer (P2P) networks and GRIDs, is becoming increasingly popular, thereby creating higher data availability and performance demands on WAN applications than ever before.

Incidentally, replication has been traditionally deployed as a means of maximizing both data availability and performance. However, we contend that replication schemes for traditional environments (e.g., clusters) [12, 16, 20] do *not* adequately address the demanding requirements of WAN environments. Issues such as node heterogeneity (in terms of processing capacity and available disk space for storing replicas), significant variations in bandwidth, lack of centralized control, lack of global knowledge, distributive ownership and scalability make replication in WAN environments significantly more challenging than in the case of traditional domains. In this regard, several research efforts [9, 19, 7, 15, 5, 18, 17, 13, 11] have been made for addressing replication in WAN environments.

However, many open questions still remain in this research area. Interestingly, similar fundamental issues typically arise when dealing with replication in different types of WAN environments. This paper examines two representative WAN environments, namely P2P systems and GRIDs. The main focus of this work is to discuss open research issues concerning replication in these two environments as well as our perspectives on these issues. Hence, we have just referred to existing replication schemes for WAN environments briefly without delving into specific details and as such, this paper is not intended to be a survey. Moreover, note that the terms 'nodes' and 'peers' have been used interchangeably throughout this paper.

The remainder of this paper is organized as follows. Section 2 briefly discusses some background information concerning replication in P2P/GRID systems, while Section 3 presents our contributions in this area. Section 4 discusses open research issues and our perspectives on these issues. Finally, Section 5 summarizes the paper.

2. Background Information

This section briefly desribes some background information concerning replication in P2P/GRID systems.

Replication in P2P networks has been discussed in [9, 6, 10]. The proposal in [9] investigates optimal replication of content in P2P systems and develops an adaptive, fully distributed algorithm which dynamically replicates content in a near-optimal manner. The works in [6, 10] discuss replication with the objective of facilitating search in P2P systems.

Replication for performing load-balancing with the objective of improving the data availability and performance of P2P systems has been investigated in [19, 7, 15]. In [19], peers are clustered based on the semantic categories of the documents contributed by them, thereby motivating the need for both intra-cluster and inter-cluster load-balancing. While intra-cluster load-balancing is achieved by exploiting meta-data, which describes associations between peer clusters and document categories or the use of routing indices, inter-cluster load-balancing is achieved by associating the document categories with clusters of nodes in a fair manner. The Cooperative File System (CFS) [7] is a P2P read-only storage system that aims at efficiency and load-balancing of file storage and retrieval via a scalable decentralized architecture. CFS servers provide a distributed hash table DHash for block storage and DHash distributes blocks for providing load-balancing and deploys replication for robustness. The work in [15], which also assumes a distributed hash table abstraction, proposes moving virtual servers from heavily loaded nodes to lightly loaded nodes for load-balancing purposes.

Replication has also been examined in GRIDs with the aim of improving data availability and reliability [5, 18, 17]. Keeping in mind the demanding I/O needs of GRID applications, the proposal in [5] discusses the design of a data GRID for data-intensive petabyte applications. The work in [18] proposes the binding of execution and storage sites together into I/O communities that participate in the wide area system. The proposal in [17] describes a data movement system (Kangaroo) which makes opportunistic use of resources (disks and networks), while hiding network storage devices behind memory and disk buffers such that background processes handle data movements. It aims at improving data availability and reliability by sacrificing consistency guarantees. For more details on GRID computing projects, interested readers may refer to the Earth Systems GRID (ESG) [1], the NASA Information Power GRID (IPG) [4], the GRID Physics Network (GriPhyN) [3] and the European DataGRID [2]. Notably, these projects deal with huge amounts of geographically distributed data (in the range of terabytes or petabytes), thereby emphasizing the importance of replication for providing high data availability.

More recently, the work in [8] discusses a light-weight middleware architecture used in the MADIS project for maintaining the consistency of replicated databases. MADIS largely makes use of basic resources provided by conventional database systems to achieve its purpose, thereby making the implementation of replica consistency protocols simple and cost-effective. MADIS enables the databases to simultaneously maintain different kinds of meta-data that are needed for different replication protocols. This allows the optimal replication protocol to be chosen on-the-fly in a 'plug-and-play' manner, depending upon the varying requirements of different applications. Note that the proposal in [8], possibly with some extensions, could be also used for replication in P2P/GRID systems.

3. Our Contributions

This section briefly summarizes our previous contributions in this area w.r.t. P2P networks and GRIDs.

3.1. P2P networks

Our proposal in [13] discusses a replication scheme for improving the data availability of an unstructured P2P system. The work assumes that every peer provides a certain amount of its disk space to the P2P system for storing the replicas of other peers' 'hot' data files. In this regard, the main contributions of the work in [13] are the proposal of a dynamic data placement strategy involving data replication for reducing the loads of the overloaded peers and the proposal of a dynamic query redirection technique which aims at reducing response times.

When an overloaded peer P_{Hot} determines that one of its 'hot' data items D needs to be replicated, P_{Hot} selects a destination peer P_{Dest} (which will store D's replica) based on the probability of P_{Dest} being online, the available disk space for replication at P_{Dest}, the load difference between P_{Hot} and P_{Dest} and the transfer time between P_{Hot} and P_{Dest}. Interestingly, since D is a 'hot' data file, D is likely to exist in the disk of at least some of the peers which had earlier queried for and downloaded D from P_{Hot}. Hence, we choose P_{Dest} from among the peers which have already downloaded D, thereby making the transfer time between P_{Hot} and P_{Dest} effectively equal to 0. This saves communication overhead significantly since files in P2P systems are typically in the range of Megabytes (for music files) and Gigabytes (for video files).

Query redirection is performed as follows. When a peer P_{Issue} issues a query Q to an overloaded peer P_{Hot} for retrieving D, P_{Hot} chooses a specific peer $P_{Redirect}$ to which Q should be redirected based on the probability of $P_{Redirect}$ being online, the load difference between P_{Hot}

and $P_{Redirect}$ and the transfer time between $P_{Redirect}$ and P_{Issue}.

3.2. Spatial Data GRID

The increasing availability of geographically distributed spatial data and the prevalence of spatial applications provides a strong motivation for designing a spatial data GRID with high data availability i.e., it allows its users to access the data of any location from *anywhere* at *any time*. Scientific applications that require virtual collaboration across the globe would benefit tremendously by deploying a spatial data GRID.

Our work in [11] has focussed on dynamic and online load-balancing in such a spatial GRID via data movement (data replication/migration[1]) for improving data availability and system performance. The main contributions of the work include envisaging the spatial GRID as comprising several clusters where each cluster is a local area network (LAN) and the proposal of a novel inter-cluster load-balancing algorithm which facilitates data availability by means of a novel and scalable dynamic data placement scheme involving data replication/migration. Observe that separation of concerns between intra-cluster and inter-cluster load-balancing issues facilitates system amenability, which is especially critical for GRIDs, which may possibly encompass hundreds of thousands of geographically distributed nodes.

In [11], we also examine trade-offs between data replication and data migration in the context of the proposed spatial GRID. For addressing variations in indexing mechanisms across the nodes of the GRID, we move the data as opposed to moving the spatial indexes. Variations in processing capacities are dealt with by normalizing the respective loads of the nodes w.r.t. their processing capacities. For handling significant differences in available disk space at different nodes in a GRID, we propose the 'pushing' of non-hot data (via migration for large-sized data and via replication for small-sized data) to large-capacity nodes as much as possible. Moreover, we advocate the replication of small-sized 'hot' data at small-capacity nodes and the movement of large-sized 'hot' data to large-capacity nodes.

4. Open questions and perspectives

This section discusses open questions concerning replication in P2P/GRID systems and our perspectives on these open questions.

[1]Unlike replication, migration implies that once 'hot' data have been transferred to a destination node, they will be **deleted** at the source node.

4.1. What does replication really mean in the P2P/GRID context?

In traditional distributed environments, nodes either completely cooperate to decide upon replication or some 'master' node decides upon replication in a centralized manner. However, in P2P/GRID systems, nodes are distributively owned, thereby implying that the level cooperation between nodes can be reasonably expected to be low at best, let alone complete cooperation. The implication is that a data item D may be replicated at a node $Dest$ subject to the condition that the owner of $Dest$ is willing to store D's replica. Hence, for P2P/GRID systems, a decision concerning replication is not necessarily definitive, instead it is just a *request* for replication. It may also be viewed as a *plan* for performing a replication, the final outcome of which depends upon the approval of $Dest$'s owner.

Additionally, in contrast with traditional distributed systems, nodes in P2P/GRID systems are allowed to delete replicas stored at themselves autonomously, without having to inform any other node concerning these deletions. Moreover, observe that in P2P systems, peers may go offline anytime without informing other peers, thereby creating the same effect as that of deletion of *all* the replicas stored at themselves for the duration of time when they are offline. Hence, even if D has been replicated at $Dest$, there are no guarantees concerning the duration of time for which D's replica will actually exist at $Dest$. In essence, replication in the context of P2P/GRID systems is typically associated with a fair amount of *uncertainty* that does *not* arise for traditional distributed systems.

4.2. Active Replication vs Passive Replication

Two modes of replication are possible in case of P2P systems, namely *active replication* and *passive replication*. Active replication implies that the peer storing a 'hot' data item D attempts to replicate D at some other peer with the intent of offloading some of its own load onto that peer. Interestingly, the implication of a data item D being 'hot' is that D must have been downloaded by several peers during recent time intervals. Now some of the peers that had downloaded D may decide to share D with other peers. We designate this phenomenon as *passive replication* because the original owner of D does not make any attempt at replicating D, but in effect, D still gets replicated.

Intuitively, if most of the peers that downloaded D decide to share D with other peers, passive replication would probably be adequate to ensure high data availability. On the other hand, if most of the peers that downloaded D decide *not* to share D, active replication would become a necessity for data availability reasons. The questions that arise here are: *What percentage of peers decide to share the data*

that they had downloaded from other peers? What is the probability of these downloading peers being online at a specific point of time? However, no large-scale studies have yet been performed on any real P2P system for answering these questions primarily due to the difficulty of collecting large-scale P2P user statistics in a privacy-preserving manner. Notably, even approximate answers to these questions would significantly facilitate our understanding of replication in the context of real P2P systems.

Observe that the notions of active and passive replication are also applicable to GRIDs, although it may be a little less challenging to determine the answers to the above questions in GRIDs than in P2P systems partly because unlike the peers in P2P systems, the nodes in GRIDs are usually dedicated (i.e., they remain available most of the time) and partly due to the fact that the collection of behavioural statistics of users can be intuitively expected to be easier for GRIDs (where several nodes may be owned by the same organization or by collaborating organizations) than for P2P systems. However, in both P2P systems as well as in GRIDs, the above questions are extremely challenging to answer and still remain open research issues.

4.3. Keeping track of replicas

In traditional distributed environments such as clusters, centralized control almost always exists for replica management purposes. Usually, all nodes periodically report to a designated 'master' node about the replicas that are currently stored at themselves, the implication being that the 'master' node has complete knowledge concerning the respective locations of replicas. In contrast, for P2P/GRID systems, 'hot' data may be aggressively replicated across hundreds of nodes in a very transitive manner and some nodes may quickly become out of reach of the primary copy owner. Given the lack of centralized control in such systems coupled with the sheer scale and inherent dynamism, every node only has *incomplete knowledge* concerning the locations of replicas. The issue that arises here is: *How can we effectively keep track of replicas in a highly dynamic and large-scale environment with incomplete knowledge?* The complexity of this issue is further exacerbated by the fact that even the incomplete knowledge obtained by a node may be inaccurate because nodes storing replicas may delete replicas autonomously.

4.4. Determination of the number of replicas

If a node N storing a 'hot' data item D knows the number of existing replicas for D, N can make an informed decision concerning whether it is necessary to create more replicas for D. However, determination of the number of replicas corresponding to a 'hot' data item becomes extremely challenging in P2P/GRID systems because 'hot' data items may be replicated not only by the primary copy owner, but also by other nodes which store the replicas of these data items. When nodes, other than the primary copy owner, create new replicas for these 'hot' data items, they typically do not inform the primary copy owner concerning these new replicas. This may be attributed to the fact that 'hot' files being replicated aggressively across the WAN, it is not always practically feasible to determine the primary copy owner. Additionally, the possibility of nodes autonomously deleting replicas stored at themselves makes it even more difficult to determine the number of existing replicas for a 'hot' data item at a specific point of time.

In our opinion, trying to roughly *estimate* the existing number of replicas corresponding to a given 'hot' data item would be more practically useful than attempting to determine the *exact* number of replicas. Estimation of the number of replicas in P2P/GRID systems still remains a question open to further research and investigation.

4.5. Replication of relatively unpopular data

Replication should improve the availability of data, irrespective of whether the data is popular (i.e., 'hot' data) or relatively unpopular. Understandably, considerable amount of research focus has been directed towards replication of popular data because majority of the user population is interested in the popular data. As a result, replication of relatively unpopular data has received little or no attention. However, for satisfying a minority of the user population who are interested in unpopular data, each and every unpopular data item should also be kept available via replication at least at some of the nodes.

Notably, many replicas of popular data typically exist in P2P/GRID systems due to passive replication, hence popular data can usually be found within the specified hops-to-live of a given query, irrespective of where the query is being issued from. This suggests that it could be beneficial to replicate unpopular data k hops apart, where k is the hops-to-live for the given P2P/GRID system. However, trade-offs between the benefit of replicating unpopular data and the disk space required for storing such data, and the determination of nodes, where the unpopular data should be stored so that they can be found irrespective of the node from which the query is issued, still remain open research issues.

4.6. Sharing complex data types in P2P systems

Currently, P2P systems are primarily used for sharing music and video files. The question which arises is: *Should a powerful computing paradigm such as P2P be limited to just file-sharing applications?* Given that P2P systems

provide a large-scale and cost-effective mechanism for data sharing in general, more complex data types can also be shared via P2P systems. Understandably, replication associated with more complex data types in P2P systems may be significantly more complicated than just replication of files. In this regard, our work in [14] has investigated the sharing of spatial data in P2P systems primarily from the perspective of indexing. Notably, overlaps between spatial objects and the inherent complexity of spatial queries can be expected to increase the complexity of replication schemes significantly.

Incidentally, P2P environments are inherently untrustworthy, thereby indicating that the sharing of sensitive user data (e.g., credit card numbers, users' medical data) in P2P systems may *not* be practically feasible, unless some measures concerning trust, privacy and accountability are introduced for P2P systems. Some recent works [21] have examined these issues for P2P systems, but it is also important to note that introducing accountability in P2P systems runs counter to the very principles of anonymity, non-accountability and freedom that P2P systems thrive on. In essence, while non-sensitive data is highly likely to be shared in P2P systems in the future, the sharing of sensitive data in P2P systems still remains a debatable issue.

4.7. Legal issues in P2P systems

Unfortunately, legal systems all over the world have not been able to keep pace with new technologies such as P2P systems. The recent spate of court proceedings between P2P systems and music corporations related to the sharing of music files in P2P systems has highlighted several 'grey areas' in current laws when applied to P2P file-sharing. This has serious implications for replication in P2P systems. For example, if user X stores the replica of user Y's file F, can legal action be taken against X by a music corporation in case the copyright of F belongs to the music corporation? In other words, how would X know whether storing a replica constitutes a copyright infringement on his part?

Moreover, P2P systems transcend geographical and political boundaries. This further exacerbates X's problem of determining whether he is acting in accordance with the law in storing a particular replica in case the owner of the original data is governed by a different set of laws in a different country. Additionally, given the fast proliferation of replicas in P2P systems, determination of the rightful owner of the original data item is *not* practically feasible in practice.

5. Conclusion

This paper has examined effective replication in two representative WAN environments, namely P2P/GRID systems. Open research issues and perspectives have been discussed with the objective of soliciting contributions in this area from academia as well as from industry, the final aim being to ensure high data availability in WAN applications.

References

[1] Earth Systems GRID. *http://www.earthsystemgrid.org/*.

[2] European DataGRID. *http://eu-datagrid.web.cern.ch/eu-datagrid/*.

[3] GriPhyN Project. *http://www.griphyn.org/index.php*.

[4] NASA IPG. *http://www.ipg.nasa.gov/*.

[5] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The data GRID: Towards an architecture for the distributed management and analysis of large scientific datasets. *Proc. Network Storage Symposium*, 1999.

[6] E. Cohen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. *Proc. ACM SIGCOMM*, 2002.

[7] F. Dabek, M. Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-area cooperative storage with CFS. *Proc. SOSP*, 2001.

[8] L. Irun-Briz, H. Decker, and R. de Juan-Marin et al. MADIS: A slim middleware for database replication. *Proc. Euro-Par*, 2005.

[9] J. Kangasharju, K. W. Ross, and D. A. Turner. Optimal content replication in P2P communities. *Manuscript*, 2002.

[10] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured Peer-to-Peer networks. *Proc. ACM ICS*, 2002.

[11] A. Mondal, K. Goda, and M. Kitsuregawa. Effective load-balancing via migration and replication in spatial GRIDs. *Proc. DEXA*, 2003.

[12] A. Mondal, M. Kitsuregawa, B. Ooi, and K. Tan. R-tree-based data migration and self-tuning strategies in shared-nothing spatial databases. *Proc. ACM GIS*, 2001.

[13] A. Mondal, Y. Lifu, and M. Kitsuregawa. On improving the performance dependability of unstructured P2P systems via replication. *Proc. DEXA*, 2004.

[14] A. Mondal, Y. Lifu, and M. Kitsuregawa. P2PR-tree: An R-tree-based spatial index for Peer-to-Peer environments. *Proc. P2PDB*, 2004.

[15] A. Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica. Load balancing in structured P2P systems. *In Proc.IPTPS*, 2003.

[16] P. Scheuermann, G. Weikum, and P. Zabback. Adaptive load balancing in disk arrays. *Proc. Foundations of Data Organization and Algorithm*, 1993.

[17] D. Thain, J. Basney, S. Son, and M. Livny. The Kangaroo approach to data movement on the GRID. *Proc. HPDC*, 2001.

[18] D. Thain, J. Bent, A. Arpaci-Dusseau, R. Arpaci-Dusseau, and M. Livny. Gathering at the well: Creating communities for GRID I/O. *Proc. SC*, 2001.

[19] P. Triantafillou, C. Xiruhaki, M. Koubarakis, and N. Ntarmo. Towards high performance Peer-to-Peer content and resource sharing systems. *Proc. CIDR*, 2003.

[20] G. Weikum, P. Zabback, and P. Scheuermann. Dynamic file allocation in disk arrays. *Proc. ACM SIGMOD*, 1991.

[21] L. Xiong and L. Liu. Peertrust: Supporting reputation-based trust in peer-to-peer communities. *Proc. IEEE TKDE Special Issue on P2P Data Management*, 2004.