A multi-agent algorithm to improve content management in CDN networks

Agostino Forestiero and Carlo Mastroianni

CNR - ICAR
Via Pietro Bucci, 41C
87036 Rende (CS), Italy
Email: {forestiero,mastroianni}@icar.cnr.it

Abstract. An effective solution to delivery static contents are the Content Delivery Networks (CDNs). However, when the network size increases, they show limits and weaknesses related to their size, dynamic nature, and due to the centralized/heirarchical algorithms used for their management. Decentralized algorithms and protocols can be usefully employed to improve their efficiency. A bio-inspired algorithm that improves the performance of CDNs by means of a logical organization of contents is presented in this paper. Self-organizing ant-inspired agents move and organize the metadata describing the content among the CDN servers, which are interconnected in a peer-to-peer fashion, so as to improve discovery operations. Experimental results confirm the effectiveness of the adopted approach.

Keywords: Content Delivery Networks; Bio-inspired; Peer to Peer

1 Introduction

Content Delivery Networks are an efficient alternative to centralized storage for the delivery of static and dynamic content, such as video on-demand, TV broadcasts, media streaming services, pay-per-use software, pay-per-download music, etc. Content replication and distribution is adopted by CDNs to improve the performance of Internet-based content delivery in terms of response time and accessibility. Clusters of surrogate servers, located at the network edge, are maintained and geographically distributed in order to put content as close as possible to the users.

Nowadays, many aspects of Content Networks have been improved in aspects such as the available content, the number of hosts and servers, the kind and the number of the users and the efficiency of real time services. The best surrogate servers - that store copies of the content - are chosen to satisfy user requests. Hence, a system and a set of mechanisms able to provide contents and services in a scalable manner need to be offered. With the explosion of social networks and P2P technologies, the amount of content has increased hugely, as well as the exploitation of the Cloud Computing paradigm, in which numerous servers located in the "Clouds" manage the content and the services. However, to perform retrieval or access operations, current applications that create, modify and manage the content, and actively place it at appropriate locations, are often insufficient. Small- or medium-sized networks can be acceptably tackled with a centralized approach. However, the CDN paradigm shows its limits in large-scale and dynamic systems. Decentralized algorithms and protocols, such as peer-to-peer (P2P) and multi agent systems, can be useful to deal with new technologies and complex paradigms [7][9].

In this paper an algorithm that exploits nature-inspired agents to organize the content in Content Delivery Networks, is presented. This approach was first introduced in [4], where a high-level description was given. Here the approach is described in more details, specifically regarding the content discovery procedure, and performance results are presented and discussed. Metadata documents that describe the content are moved and logically organized by the agents to improve information retrieval operations. In our approach, metadata documents are indexed by binary strings, obtained as the result of the application of a locality preserving hash function, which maps similar resources into similar binary strings. For example, each bit of the string may represent the absence or presence of a given characteristic of an offered service. Agents move across the CDN network through the peer-to-peer interconnections moving the metadata documents. Similar metadata, representing similar resources, are located into the same or in neighbor hosts/servers. The assignment of metadata documents to CDN servers is self-organizing and driven by probabilistic operations, and easily adapts to the dynamic conditions of the network.

The logical reorganization induced by the operations of mobile agents allows to exploit the benefits of structured and unstructured approaches adopted in peer to peer systems. The logical reorganization of the metadata documents improves the rapidity and effectiveness of discovery operations, and enables the execution of range queries, i.e., requests of content that matches some specified features. In fact, thanks to the features of the hash function, the metadata strings that differ only by a few bits will be located in neighboring regions. To measure the similarity between two binary strings the Hamming distance or the cosine of the angle between the related vectors is used.

The rest of the paper is organized as follows: Section 2 discusses related work, Section 3 describes how the bio-inspired agents replicate and logically reorganize metadata documents on the CDN network and Section 4 describes the discovery algorithm that exploits the metadata reorganization to perform simple and range queries. Finally, in Section 5 the performance analysis of the algorithm is reported.

2 Related Works

Approaches that combine CDNs and P2P methodologies have been analyzed by several studies: [12] [11] proposed the use of P2P to deliver multimedia content; [14] [10] exploit P2P overlays for surrogate cooperation while leaving the clients as regular non-cooperative entities; in [16], a collaboration between clients is proposed, but clients cannot receive data at the same time from different sources, such as from the peering community and CDN entities. Some interesting works that propose the adoption of P2P and multi agent systems in Content Delivery Networks, are collected in [7] and [8].

The dynamic nature of today's networks and the large variety of the resources make management and discovery operations very complex. Administrative bottlenecks and low scalability of centralized systems are becoming unbearable. Innovative approaches need to have properties such as self-organization, decentralization and adaptivity. Erdil et al. in [3] outline the requirements and properties of self organizing grids, where reorganization of resources and adaptive dissemination of information are applied to facilitate discovery operations. A class of agent systems which aim to solve very complex problems by imitating the behavior of some species of ants was introduced in [1]. In [6] and [5], the performance of discovery operations is improved through the creation of Grid regions specialized in specific classes of resources, and [15] proposes a decentralized scheme to tune the activity of a single agent. These systems are positioned along a research avenue whose objective is to devise possible applications of ant algorithms [1] [2]. In [13], a tree-based ant colony algorithm was proposed to support large-scale Internet-based live video streaming broadcast in CDNs. Here, differently from the traditional solutions adopted to find a path towards a target resource, an algorithm is introduced to integrate and optimize multicast trees into the CDN network.

3 Algorithm for Metadata Reorganization

The approach presented here is composed by two main algorithms, an algorithm for metadata reorganization and another for the discovery of metadata documents. The main purpose of the first algorithm is to disseminate metadata over the CDN network and at the same time achieve a logical organization of content by spatially sorting the metadata in accordance to the corresponding indexes, or binary strings. Operations of nature-inspired agents are profitably exploited to reallocate the metadata. Agents move among CDN servers, or hosts, performing simple operations. When an agent arrives to a host and it does not carry any metadata document, it decides whether or not to pick one or more documents stored in the current host. When a loaded agent arrives to a new host, it decides whether or not to leave one or more metadata documents in the local host. Probability functions drive agents' decisions. The probability functions are based on a similarity function, that is:

$$sim(\bar{m}, R) = \frac{1}{N_m} \sum_{m \epsilon R} 1 - \frac{Ham(m, \bar{m})}{dim} \tag{1}$$

This function measures the similarity of a metadata binary string $\bar{m}$ with all the other strings located in the local region. The length of string is assumed to be equal to dim. The local region R for each server s includes s and all the host reachable from s in a number of hops h. The value of h is set to 1 unless otherwise stated. N_m is the overall number of metadata documents located in R, while $Ham(m, \bar{m})$ is the Hamming distance between a metadata document m and $\bar{m}$. The value of the function *sim* ranges between 0 and 1. The probability of picking a metadata document from a server must be inversely proportional to the similarity function *sim*. On the other hand, the probability function of dropping a metadata must be directly proportional to the similarity function *sim*. In this way, an agent tends to pick metadata documents that are dissimilar to the other documents stored locally, and will move and drop them to other regions where more similar documents are stored, so improving the spatial reorganization of metadata.

According to these considerations, the probability functions of picking a metadata P_1 and the probability function of dropping a metadata P_2, are:

$$P_1 = \left(\frac{k1}{k1 + sim(\bar{m}, R)} \right)^2 \quad (2)$$

$$P_2 = \left(\frac{sim(\bar{m}, R)}{k2 + sim(\bar{m}, R)} \right)^2 \quad (3)$$

The degree of similarity among metadata documents can be tuned through the parameters *k1* and *k2*, which have values comprised between 0 and 1, and in this work have been set, respectively, to 0.1 and 0.3, as in [1]. The flowchart showed in Figure 1 gives a high-level description of the algorithm performed by mobile agents. Cyclically, the agents perform a given number of hops among servers and, when they get to a server, they decide which probability function they must use, based on their state. If the agent does not carry metadata it computes P_1, otherwise it computes P_2.

The effectiveness of the algorithm has been evaluated by defining the spatial uniformity function, i.e. the average homogeneity of metadata documents stored in neighbor servers. The uniformity U_s of the documents stored in a local region centered in the server s is defined as:

$$U_s = dim - Avg_{m_1, m_2 \epsilon R} Ham(m_1, m_2) \quad (4)$$

where m_1 and m_2 are two metadata documents stored in the local region R. The value of the global uniformity U is obtained by averaging the values of U_s over all the servers of the network.

Simulation tests showed that the uniformity function is better increased if each agent works in two operational modes, *copy* and *move*. In the first phase of its life, an agent is required to *copy* the metadata that it picks from a server, but when it realizes from its own activeness that the reorganization process is at an advanced stage, it begins simply to *move* metadata from one host to another, without creating new replicas. In fact, the *copy* mode cannot be maintained for a long time, since eventually every host would store a very large number of metadata of all types, thus weakening the efficacy of spatial reorganization. The

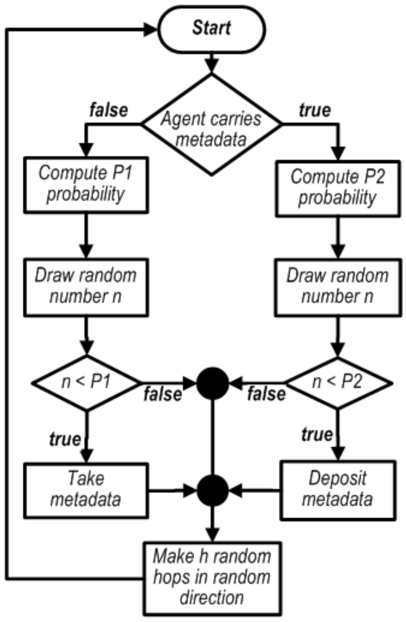

Fig. 1. The algorithm for metadata reorganization performed by agents.

algorithm is effective only if each agent, after replicating a number of metadata, switches from *copy* to *move*.

A self-organization approach, in some sense similar to that used in [15], enables each agent to tune its activeness, in our case to perform a mode switch, only on the basis of local information. Our approach is inspired by the observation that agents perform more operations when the system disorder is high, because metadata are distributed randomly, but operation frequency gradually decreases as metadata documents are properly reorganized. The reason for this is that the values of the functions P_1 and P_2, defined in expressions (2) and (3), decrease as metadata documents are correctly replaced and reorganized on the network.

With a mechanism inspired by ants and other insects, each agent maintains a *pheromone base* (a real value) and increases it when its activeness tends to decrease, which means that the disorder level has significantly decreased: the agent switches to the *move* mode as soon as the pheromone level exceeds a defined threshold T_h. In particular, precisely each 500 time units, each agent counts the number of times that it has evaluated the P_1 and P_2 probability functions, $N_{attempts}$, and the number of times that it has actually performed pick and drop operations, $N_{operations}$. At the end of each time interval, the agent makes a deposit into its pheromone base, which is inversely proportional to the fraction of performed operations. An evaporation mechanism is used to give a greater weight to the recent behavior of the agent. Specifically, at the end of the i-th time interval, the pheromone level Φ_i is computed with the following expression:

$$\Phi_i = Ev \cdot \Phi_{i-1} + \left(1 - \frac{N_{operations}}{N_{attempts}}\right) \qquad (5)$$

The evaporation rate E_v is set to 0.9 [15], whereas ϕ_i is the amount of pheromone deposited in the last time interval. The pheromone level can assume values comprised between 0 and 10: the superior limit can be obtained by equalizing Φ_i to Φ_{i-1} and setting ϕ_i to 1. As soon as the pheromone level exceeds the threshold T_h (whose value is set to 9 in this work), the agent switches its mode from *copy* to *move*.

4 Algorithm for Metadata Discovery

The reorganization and sorting of metadata can be exploited by a discovery algorithm that allows users to find the resources or services that they need for their applications. In a CDN, users often need to locate resources with given characteristics and, after retrieving a number of them, they can choose the resources that best fit their needs. Accordingly, a query message is issued by a user to search for "target metadata", that is, for metadata documents having a given value of their binary index. The query is forwarded through the CDN network, hop by hop, so as to discover as many target metadata documents as possible. Thanks to the spatial sorting of metadata achieved by ant-based agents, the discovery procedure can be simply managed by forwarding the query, at each step, towards the "best neighbor", that is, the neighbor that maximizes the similarity between the metadata stored locally and the target metadata.

Each CDN server computes a "centroid" metadata. This metadata is a vector of *dim* real numbers comprised between 0 and 1, and is obtained by averaging all the local metadata indexes. Specifically, the value of each centroid element is calculated by averaging the values of the bits, in the same position, of all the metadata stored in the local peer. For example, the centroid metadata of a server that maintains the three metadata [1,0,0], [1,0,0] and [0,1,0] is a metadata having an index [0.67,0.33,0]. Before forwarding a query, the cosine of the angle between the query target metadata, and the centroids of all the neighbors, is computed. This value gives a hint about how much the metadata of the neighbors are similar to the target metadata. Thereafter, the query is forwarded to the neighbor that maximizes this cosine similarity index. At the next step, the target server will do the same so that, step by step, the query approaches a region of the network where it is more and more likely to discover several useful results, that is, target metadata. The search is terminated whenever it is no longer advantageous to forward the query, that is, when the best neighbor is not better than the server where the query has arrived so far. At this point, a queryHit message is issued and returns to the requesting host by following the same path, and collecting on its way all the results that it finds.

The efficient resolution of range queries is a fundamental requirement of CDN systems. A range query is defined as a query in which the bit vector of the target metadata contains one or more wildcard bits, that can assume either the 0 or the 1 value. This means that a range query can return descriptors having 2^W possible values, if W is the number of wildcard bits. Of course, the assignment of indexes to metadata must assure that the bit vectors that correspond to similar resources are also similar to one another. This can be done without losing generality by using the binary Gray code, in which two successive indexes always differ by only one bit. To select the best neighbor, the cosine similarity is still computed between the target metadata and the centroid metadata of all the neighbors, but this time these indexes are preprocessed by discarding the bits that are defined as wildcards in the target metadata. Therefore, only the centroid bits that correspond to valued bits in the target metadata are useful to drive the query message. As for simple queries, a range query terminates its journey when it is no more possible to find a better neighbor. The queryHit message will come back and collect all the metadata that match the range query. Range queries are not able to discover all the resources that would be found with the corresponding number of simple queries, but provide an efficient way to discover – in just one shot – much more results than a simple query. It can be concluded that the resource management of the algorithm actually facilitates this objective, as showed in Section 5.

5 Experimental results

An event-based simulator, written in Java, was implemented to evaluate the performance of the algorithm. Prior to the numerical analysis, a graphical description of the behavior of the algorithm, for the case in which *dim* is set to 3, is given in Figure 2. Here, 2,500 CDN servers are arranged in a grid and each metadata is associated to a RGB color.

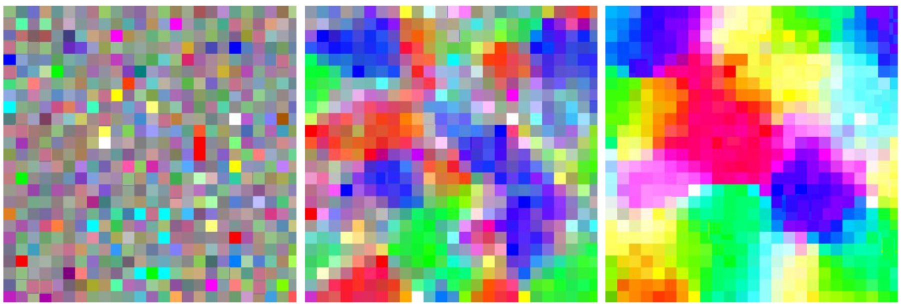

Fig. 2. Snapshots of the system showing the reorganization of metadata documents indexed by 3 bits, represented by RGB colors. The snapshots are taken when the process starts, in an intermediate state and in a steady state.

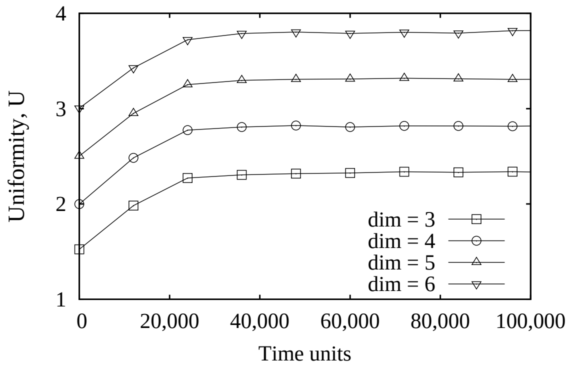

Fig. 3. Uniformity of the whole network when the number of bits of the binary string representing the content ranges from 3 to 6.

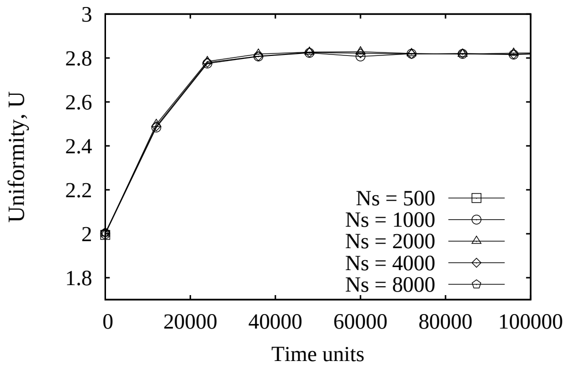

Fig. 4. Uniformity, vs. time, for different values of the number of servers.

Each server is visualized by means of the RGB color of the metadata with the highest number of elements placed in it. Three snapshots of the network are depicted: the first is taken when the process is initiated (time units = 0), the second is taken 10,000 time units later, and the third snapshot is taken in a quite steady situation, 100,000 time units after the process start. Notice that similar metadata are located in the same region and that the color changes gradually, which proves the spatial sorting of metadata on the network. Figure 3 shows the value of the overall uniformity — defined in Section 3 – when the number of bits of the metadata describing the content,*dim*, is varied. We can see that the logical reorganization is obtained independently of the number of bits.

To confirm the scalability nature of the algorithm, which derives from its decentralized and self-organizing characteristics, its behavior with different numbers of servers Ns, between 500 and 8000, was analyzed and reported in Figure 4. It is noticed that the size of the network has no detectable effect on the overall uniformity index.

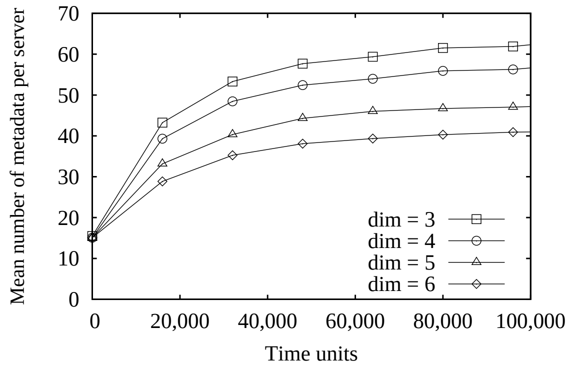

Fig. 5. Mean number of metadata documents handled by a server when the length of binary strings ranges from 3 to 6 bits.

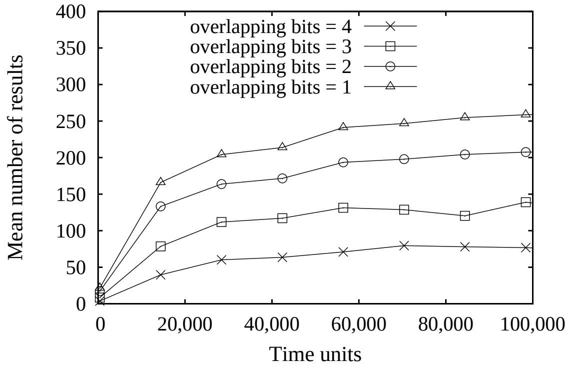

Fig. 6. Mean number of results collected by a range query when the length of the binary string representing the content is set to 4 and the number of overlapping bits ranges from 1 to 4.

Figure 5 reports the average number of metadata documents that are maintained by a server at a given time. Indeed, one of the main objectives of the algorithm is the replication and dissemination of metadata. This objective is achieved: the number of metadata documents maintained by a server increases from an initial value of about 15 (equal to the average number of resources published by a server) to much higher values; the trend of this value undergoes a transient phase, then it becomes stabilized, even if with some fluctuations.

To evaluate the effectiveness of range queries, the length of binary string – *dim* – was set to 4, and queries are issued in which some bits of the target binary string are wildcard bits, while other bits are specified. The latter are called *overlapping* bits in the following. The average number of results collected by a range query, when the number of overlapping bits ranges between 1 and 4, is shown in Figure 6. It appears that, in a steady situation, the number of results increases with the number of overlapping bits, as each additional overlapping bit doubles the number of admissible results. Range queries are not able to discover all the results that would be obtained by issuing a simple query for each admissible value of the target binary index. However, range queries provide an efficient way to discover – in just one shot – much many results than a single query.

6 Conclusions

This paper presents a nature-inspired approach to build a P2P information system for CDNs. Thanks to its swarm intelligence characteristics, the proposed algorithm features fully decentralization, adaptivity and self-organization. Ant-inspired agents move and logically reorganize the metadata documents representing the content or the services. Agent operations are driven by simple probability functions that are evaluated when agents get to a new server. In this way, similar metadata documents representing similar contents are placed in the same region, that is on neighbor CDN servers. Performance analysis, achieved through event-based simulation, confirms the effectiveness of the approach and the increased efficiency of discovery operations – specifically of range queries – obtained thanks to the logical reorganization of metadata documents.

References

1. Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems*, volume 4. Oxford university press New York, 1999.
2. Marco Dorigo, Eric Bonabeau, and Guy Theraulaz. Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16(8):851–871, 2000.
3. Deger Cenk Erdil, Michael J Lewis, and Nael B Abu-Ghazaleh. Adaptive approach to information dissemination in self-organizing grids. In *Autonomic and Autonomous Systems, 2006. ICAS'06. 2006 International Conference on*, pages 55–55. IEEE, 2006.
4. Agostino Forestiero. Self organization in content delivery networks. In *Parallel and Distributed Processing with Applications (ISPA), 2012 IEEE 10th International Symposium on*, pages 851–852, July 2012.
5. Agostino Forestiero, Carlo Mastroianni, and Giandomenico Spezzano. Reorganization and discovery of grid information with epidemic tuning. *Future Generation Computer Systems*, 24(8):788–797, 2008.
6. Agostino Forestiero, Carlo Mastroianni, and Giandomenico Spezzano. So-grid: A self-organizing grid featuring bio-inspired algorithms. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 3(2):5, 2008.
7. Giancarlo Fortino and Carlo Mastroianni. Enhancing content networks with p2p, grid and agent technologies. *Future Generation Computer Systems*, 24(3):177–179, 2008.
8. Giancarlo Fortino and Carlo Mastroianni. Next generation content networks. *Journal on Network and Computing Applications*, 32(5):941–942, 2009.
9. Giancarlo Fortino and Wilma Russo. Using p2p, grid and agent technologies for the development of content distribution networks. *Future Generation Computer Systems*, 24(3):180–190, 2008.

10. Zhang Guomin, Xing Changyou, and Chen Ming. A distributed multimedia cdn model with p2p architecture. In *Communications and Information Technologies, 2006. ISCIT'06. International Symposium on*, pages 152–156. IEEE, 2006.
11. Cheng Huang, Angela Wang, Jin Li, and Keith W Ross. Understanding hybrid cdn-p2p: why limelight needs its own red swoosh. In *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 75–80. ACM, 2008.
12. Shi Kang and Hao Yin. A hybrid cdn-p2p system for video-on-demand. In *Future Networks, 2010. ICFN'10. Second International Conference on*, pages 309–313. IEEE, 2010.
13. Gang Liu, Hua Wang, and Hong Zhang. An ant colony optimization algorithm for overlay backbone multicast routing in content delivery networks. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 1878–1882. IEEE, 2012.
14. Jaison Paul Mulerikkal and Ibrahim Khalil. An architecture for distributed content delivery network. In *Networks, 2007. ICON 2007. 15th IEEE International Conference on*, pages 359–364. IEEE, 2007.
15. H Van Dyke Parunak, Sven A Brueckner, Robert Matthews, and John Sauter. Pheromone learning for self-organizing agents. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(3):316–326, 2005.
16. Dongyan Xu, Sunil Suresh Kulkarni, Catherine Rosenberg, and Heung-Keung Chai. Analysis of a cdn–p2p hybrid architecture for cost-effective streaming media distribution. *Multimedia Systems*, 11(4):383–399, 2006.