MediaDART: a decentralized framework for sharing multimedia content

Maurizio Agelli, Orlando Murru
CRS4
{agelli, orlando}@crs4.it

Abstract

This paper provides an overview of MediaDART, a framework for building online services for distributing and sharing digital media. Inspired by the participative model of Web 2.0, MediaDART relies on a scalable and decentralized architecture that can grow with the contribution of users. The architecture is based on an arbitrary number of nodes interconnected through a p2p network implementing a distributed hash table (DHT). The DHT provides resource storage and parallel resource processing for operations of feature extraction, adaptation and composition. MediaDART adopts application-level multicast based on distribution trees for delivery in streaming and implements algorithms to dynamically replicate resources across the network. The framework allows content description through user-defined tags. Tools for personalized content retrieval based on recommendation algorithms and user profiling are included too. This paper also describes two prototype applications and outlines further work.

1. Introduction

Technological evolution is radically changing many distribution models, such as the television and music industry, that up to a decade ago were considered well-established. The spreading of broadband access technologies, the availability of more efficient compression techniques and the ever decreasing cost of storage are the most evident aspects (but not the only) of this evolution. Total Internet traffic is expected to double every two years through 2011, mostly driven by the delivery of video, whose impact on IP traffic composition will grow from 20% of 2005 to 60% of 2011 [1].

In addition, Web 2.0 represents a paradigm shift in the way users approach the web. Users are no longer passive consumers of published content, but become involved both implicitly and explicitly in adding value to an application, by providing their own content in an “architecture of participation” [2]. We are witnessing a proliferation of user-generated digital media that are the sole content source for services like YouTube [3], Flickr [4], MySpace [5], and many others. However, the path to the age of content abundance, where the above depicted scenario seems to lead, is not without challenges and issues.

Firstly, delivering high-quality streaming video to an ever-increasing population of users is very demanding in terms of network capacity. For example, the fastest optical link available today (OC-768) has a transmission speed of about 40Gbit/s, nearly sufficient to stream standard resolution video (~1Mbit/s in H.264) to 40.000 viewers, or to deliver in download (assuming 1 download per day) a popular video show to 1 million households [6]. This highlights that point-to-point data transport over the Internet is not suitable for delivering large amounts of video content. Neither is multicast a practicable solution, for the simple reason that network administrators generally do not enable multicast traffic across their routers, to avoid the associated costs [7]. Nowadays, Content Delivery Networks (CDNs) provide an effective solution for high performance content delivery. CDNs are trusted overlay networks based on thousands of servers that, strategically located in the nodes of the distribution tree, act as content caches. However, CDNs have costs per GB that, for a 100TB/month account, may vary between $0.15 and $0.24, raising up to $2.00 for a 1TB/month account [8]. This means that only organizations with adequate financial resources can afford adopting CDNs for delivering their content. Another technology for distributing media on a large scale is based on peer-to-peer (p2p) networks, that were made very popular by file sharing applications like Napster [9], Gnutella [10], BitTorrent [11]. By moving all distribution costs to the last mile and all caching costs to the clients, p2p is by far the least expensive solution for delivering video, involving a cost per GB (download) of $0.0027 [12].

Secondly, the proliferation of multimedia content raises the problem of indexing and quering non-textual information [13]. Recommendation algorithms, widely used on e-commerce sites, can play a key role in digital media distribution, either by providing a targeted offer of video or music content or by creating a totally personalized media experience. In particular, item-to-item recommendation systems seem the most suitable in terms of processing time and scalability [14].

Finally, providing effective solutions of intellectual property rights protection is important for any content distribution service that intends to turn profitable. Authors' common concern has always been about being able to distribute their content while being able to retain control of it. However, well established market players have often used closed DRM technologies to restrict competition and maintain their predominant position. In theory, interoperable DRM technologies can encourage competition by lowering the threshold to content distribution and create a profitable niche market [15]. Experience has shown that any working DRM system is an end-to-end, closed solution, aimed to cope in the shortest possible time with securities breaches rather than to guarantee interoperability. Apple's iTunes [16] and Sony's Connect [17] fall in this category. Microsoft, for example, after a failed attempt to develop an openly licensed DRM solution (PlayforSure [18]) has reverted to offering a closed, end-to-end system with Zune [19]. DRM has had the side-effect of making life for the users harder, denying functionalities and features that are perfectly legal, on the basis that they could be used to infringe copyrights. This does not make much sense in a world in which over 90% of the music is sold on CDs with no form of copy protection whatsoever [20].

This paper provides an overview of MediaDART (Media Distribution, Adaptation and Retrieval Toolset), a framework for managing and distributing digital media. In particular, this paper explains how MediaDART addresses the above depicted issues and challenges.

2. Framework overview

MediaDART is a framework for building digital media distribution and sharing services. Inspired by the participative model of Web 2.0, MediaDART relies on a scalable and decentralized architecture that can grow and evolve with the contribution of users themselves.

MediaDART is focused on three main functional domains: *Distributed Storage, Content Management, Multimedia Processing.*

Distributed Storage encompasses all functional components related to delivery. This includes services that allow a user to upload, download, and stream multimedia content, striving to efficiently use the available storage and bandwidth resources. MediaDART adopts p2p technologies and implements algorithms to dynamically replicate across the p2p network the most-requested resources.

Content Management is related to information retrieval and includes functions for modeling, organizing, retrieving and personalizing content. MediaDART provides the following services: (1) an item-based recommender system; (2) a user profiler, based on clustering on the output of a text based categorization system; (3) a user-defined taxonomy system; (4) a MediaDART object repository, which allows the organizing of content using a representation model that can be configured to meet the needs of different application domains and which acts as entry point to the system; (5) a tagging system that allows a content classification based on folksonomies [21].

Multimedia Processing groups all functions that are related to resource processing. This includes adapting a content to a new format, creating new content by aggregating available resources, extracting features to be used in a variety of settings.

As far as DRM is concerned, MediaDART adopts a lightweight approach. Since adding copy protection features to an open framework like MediaDART would be technically challenging and would limit the features that could be offered, we chose to concentrate our efforts on copyright protection, rather than copying protection. MediaDART provides support for Creative Commons [22] licenses. The possibility to embed license information in the resource using an invisible watermark is also provided. Apart from this, MediaDART does not implement any stronger form of DRM.

3. Architecture

MediaDART is based on a service-oriented architecture, that can be conceptually split into two tiers (fig. 1). The M-Hub contains the application and all metadata-related services (repository, tagging system, recommender). The M-Net is dedicated to resource storage, processing and delivery.

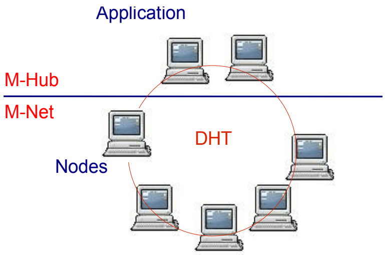

Figure 1. MediaDART tiers

The M-net is distributed across an arbitrary number of nodes. A node is a single computer that is capable to provide all resource-related MediaDART services. Nodes are interconnected using a peer-to-peer (p2p) network implementing a Distributed Hash Table (DHT). A DHT is a self-organizing overlay network that performs application level routing and object location among a very large number of nodes. The p2p network does not require the intermediation of a centralized server, and has the ability to self adapt in case of node failures or when a node enters or leaves the network.

Each resource in MediaDART is described by Resource Records (fig. 2), which are pairs <*key, value*>, where key is a unique 160-bit integer and value can be a set of attributes describing the resource, such as pointers to the physical copies of the resources and usage metadata to drive replication.

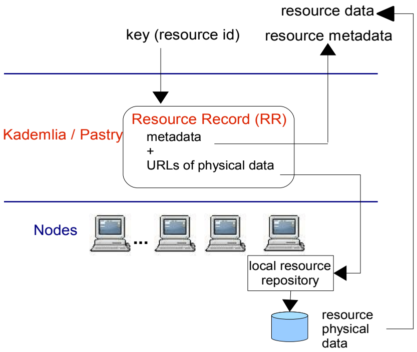

Figure 2. Resource Records

MediaDART is largely independent of the particular DHT design. However, we have three key requirements: short response time, light memory footprint, small usage of bandwidth for low churn rates in the network. MediaDART can be built atop Pastry [23] and Kademlia [24], both of which provide interesting additional features.

Pastry comes with application level services for building distribution trees and managing the persistence and replicas of resource records.

Kademlia is inherently simpler, its routing table structure makes it easy to compute global network properties from statistics extracted locally and its basic operations use an iterative, asynchronous algorithm that can be optimized to have very short response times. Moreover, its implementation is lightweight: we are exercising inner core algorithms on simulated networks of millions of nodes, and a single PC (Pentium IV, 2GB RAM) in our laboratory is able to host up to 4000 (real) Kademlia nodes, opposed to a maximum number of 60 Pastry nodes.

MediaDART nodes are used for operations of resource processing, such as extraction of features, adaptation and composition. A great deal of academic research effort has been devoted to the study of automatic annotation techniques based on the extraction of low-level features of the resources. Although tackling this kind of problems is not in the scope of MediaDART, the framework can include (as plug-ins) third-party feature extraction algorithms across the M-net. It is the responsibility of the application to define which features shall be extracted and how they shall be used.

In order to make content accessible from different usage environments, the adaptation of resources may be required. Adapting a resource implies the creation of a new resource by applying operations of transcoding, batch editing, resizing, or bit-rate modification. The composition is an operation that produces a new resource by spatially and/or temporally aggregating some available resources according to a given set of rules. The composition is particularly interesting for producing personalized content (e.g. by inserting targeted ads). MediaDART services may operate on lists of resources distributing the processing on different nodes.

Resources delivery does not require special players on the end-system. The current prototypes use Flash video and standard RTP to deliver resources to end-systems. This enhances usability of the framework, at the expense of bigger computational loads on MediaDART servers, as it may be required to re-encode streams for delivery.

Resources are moved around in MediaDART in two more cases: replication of infrequently used resources and caching of the most frequently used resources that shall be available for real time delivery. The first case guarantees the persistence in MediaDART of resources in the long tail of the usage distribution. It is managed by a Replication Agent based on the well known

BitTorrent protocol [25], adapted to use trackerless torrents and the MediaDART DHT. The Replication Agent is located in the M-hub and operates through the distribution of feedbacks on usage.

Caching is performed by using application level multicasting with distribution trees built on top of the DHT that grow with the number of simultaneous requests, and by caching resources on non-leaf nodes. The distribution tree optimizes geographical parent-child distance, which is strongly correlated with throughput and latency between Internet servers [26]. We are currently using information extracted from GeoIP [27].

MediaDART adopts an object-oriented content representation model inspired by Mpeg-21 DID [28] and based on the concept of MediaDART object: a structured set of descriptors that identify properties (metadata) and resources (data) of a given digital asset. MediaDART provides a centralized repository that makes objects persistent and allows to customize description schemes depending on the specific application needs. A MediaDART object contains a set of descriptors and a set of references to a list of associated components. Components are objects that, acting as cached copies of Resource Records, provide a structured description of resources.

Besides this top-down approach for content description, MediaDART includes a tagging system that allows describing content using a bottom-up approach based on tags arbitrarily defined by users. Although the approach based on the description scheme is semantically more rigorous, tags are more intuitive and provide a description that becomes more and more reliable as long as the community of users that participate in the tagging process grows [29]. It is up to the MediaDART applications to decide in which measure to balance these two approaches. MediaDART also provides a content retrieval service based on textual queries that operate both on descriptors and tags.

MediaDART includes a recommendation system aimed at personalized content retrieval based on user profile and actual usage history. Two different approaches have been put in place. The first starts from ratings that codify the preferences given by users to a subset of available objects. Item-based algorithms [30] allow to build a recommendation list for each user. The second approach is based on the extraction of a significant set of features from all textual information associated to objects. Users are classified into a small set of representative classes by applying clustering algorithms to these features. Finally, a collaborative filter [31] can be used to build a recommendation list for each user.

MediaDART has no centralized information about nodes, so the DHT is also used for locating nodes before invoking a service. Services are invoked through a dispatcher that, after looking up the DHT, decides which are the actual nodes where a service shall be executed. Nodes provide a Node Monitor service that can be invoked in order to get information about the node itself (average load, active services, etc.) and take dispatching decisions.

A state machine associated with every MediaDART object allows to asynchronously invocate a remote method when a condition is verified. The application can use the state machine for automating a workflow based on the services distributed over the M-net.

The MediaDART framework runs on GNU/Linux (Debian or Ubuntu).

An overview of the MediaDART architecture is outlined in fig. 3.

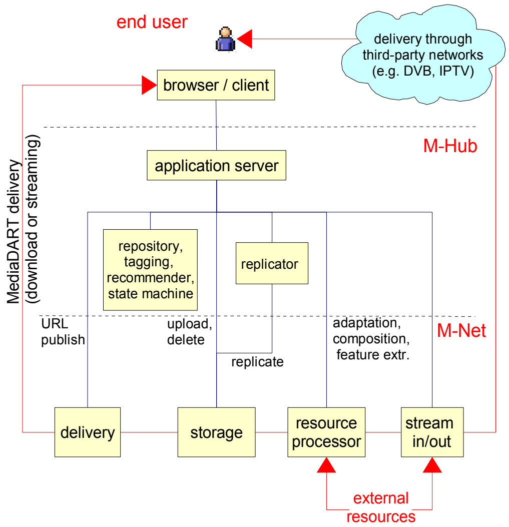

Figure 3. Architecture overview

4 - Prototypes

Two prototypes have been developed in order to verify the main functions of the MediaDART framework. The first prototype is a distributed tool for automatically creating TV programs. A client-server application allows to interactively set up composer scripts and define the scheduling of the single events within a program. The output of the composer is encoded as a Mpeg-2 TS [32] and streamed in real-time to a TV player. A single node hosted by a consumer PC has sufficient processing power to perform all required tasks in real-time: combining one

SDTV video with several images and two sliding texts from RSS feeds, re-encoding the resulting video and streaming it to the TV player.

The second prototype implements a service for collaborative media publishing on the web. The service is based on a community of users that share different types of media related to a geographical area, organizing them into different thematics (e.g. culture, sport, kitchen, sightseeings, etc.). Users can actively participate in the service by providing their own content (videos, photos, etc.), hardware resources (CPU, disk space, connectivity) or by tagging or annotating available content. Statistics on user activity are extracted in order to be made available for possible mechanisms of revenue sharing. The prototype is being used for evaluating the performance of the p2p network for delivering and processing resources.

A monitor tool has been developed in order to track in real-time the status of the nodes and to record a set of statistics about the services that are active on each node. A client emulator is used to test the system and measure the overall performance.

5 – Further work

At the time of writing this article, the MediaDART project is not yet finished and some further work is planned in several areas.

Firstly, more work is needed in the optimization of the application-level multicast tree used by MediaDART for delivering real time streams, using more complicated cost functions that take into account more variables.

Secondly, prototypes shall enter a phase of verification that will involve real users. Measures will be made during this phase in order to tune and improve algorithms in the fields of recommendation, replication of resources and dispatching of services.

6. Acknowledgments

MediaDART is being developed under the scope of the "Distributed Architecture for Semantic Search and Personalized Content Delivery" project funded by the Italian Ministry of Education, University and Research (MIUR).

7. References

[1] Cisco Systems, "Global IP Traffic Forecast and Methodology, 2006-2011", *Cisco Systems White Paper*, 2007.

[2] T. O' Reilly, "What is Web 2.0**,** Design Patterns and Business Models for the Next Generation of Software", O' Reilly, September 30th 2005, www.oreillynet.com/lpt/a/6228

[3] http://www.youtube.com

[4] http://www.flickr.com

[5] http://www.myspace.com

[6] R. Cringely, "Peering into the Future: Why P2P Is the Future of Media Distribution Even If ISPs Have Yet to Figure That Out", March 2nd 2006, http://www.pbs.org/cringely/ pulpit/2006/pulpit_20060302_000886.html

[7] R. Cringely, "The Multicast Way: Why IP Multicast Doesn't Work the Way People Keep Telling Me It Does.", March 23rd, 2006, http://www.pbs.org/cringely/pulpit/2006/pulpit_20060323_000889.html

[8] D. Rayburn, "CDN Pricing Data: What the CDNs are actually charging for delivery", BusinessofVideo.com, August 14th 2007, http://blog.streamingmedia.com/the_business_of_online_vi/2007/08/cdn-pricing-dat.html

[9] http://www.napster.com/

[10] http://www.gnutella.com/

[11] http://www.bittorrent.com/

[12] W. B. Norton, "Video Internet: The Next Wave of Massive Disruption to the U.S. Peering Ecosystem", v1.3, Equinix, 2007.

[13] M. Sahami, V. Mittal, S. Baluja, H. Rowley, "The Happy Searcher: Challenges in Web Information Retrieval", P*roceedings of the Eighth Pacific Rim Conference on Artificial Intelligence (PRICAI)*, 2004

[14] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations – Item-to-item Collaborative Filtering", IEEE Computer Society, January-February 2003.

[15] L. Chiariglione, "Balancing Protection of Intellectual Property and its Use", *Proceedings of the First International Conference of Cross Media Content for Multi-Channel Distribution (AXMEDIS ' 05)*, 2005.

[16] http://www.apple.com/itunes/

[17] http://www.connect.com/

[18] http://www.playsforsure.com/

[19] http://www.zune.net/

[20] S. Jobs, "Thoughs on Music", February 6th 2007, http://www.apple.com/hotnews/thoughtsonmusic

[21] T. Vander Wal, “Folksonomy Coinage and Definition”, February 2nd 2007, http://vanderwal.net/folksonomy.html

[22] http://www.creativecommons.com

[23] http://freepastry.rice.edu/

[24] P. Maymounkov, and D. Mazières, “Kademlia: A Peer-to-peer Information System Based on the XOR Metric”, *Proceedings of IPTPS02, Cambridge*, USA, March 2002, http://www.cs.rice.edu/Conferences/IPTPS02/109.pdf

[25] BitTorrent protocol specifications, http://www.bittorrent.org/protocol.html

[26] L. Subramanian, V.N. Padmanabhan, and R.H. Katz: “Geographic Properties of Internet Routing”, *USENIX 2002*, 2002

[27] http://www.maxmind.com/

[28] ISO/IEC 21000-2 Mpeg-21 Digital Item Declaration

[29] C. Shirky, “Ontology is Overrated: Categories, Links, and Tags”, 2005, *Clay Shirky's Writings About the Internet*, http://shirky.com/writings/ontology_overrated.html

[30] Sarwar B., Karypis G., Konstan J., Riedl J., Item-based collaborative Filtering Recommendation Algorithms, *Proceedings of the 10th international conference on World Wide Web* , Hong Kong, 2001.

[31] Blanco-Fernandez Y., Pazos-Arias J.J., Gil-Solla A., Ramos-Cabrer M., Barragans-Martinez B., Lopez-Nores M., “A Multi-Agent Open Architecture for a TV Recommender System: A Case Study using a Bayesian Strategy”, *Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering (ISMSE'04)*, Dec. 2004, pp. 178-185.

[32] ISO/IEC 13818-1 Mpeg-2 System