Journal of Network and Computer Applications 37 (2014) 127–145

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

An agent-based approach for the design and analysis of content delivery networks

Giancarlo Fortino*, Wilma Russo, Marino Vaccaro

DEIS—University of Calabria, Via P. Bucci, 87036 Rende (CS), Italy

ARTICLE INFO

Article history:
Received 6 June 2012
Received in revised form
14 October 2012
Accepted 19 November 2012
Available online 1 December 2012

Keywords:
Agent-based modeling and simulation
Content delivery networks
Multi-agent systems
Performance evaluation

ABSTRACT

Content delivery networks (CDNs) are currently the most exploited worldwide solution for reliable, scalable and cost-effective delivery of content over the Internet. The development and deployment of new services and components in CDNs should be carried out carefully as they involve huge investments from the CDN providers' perspective. It is therefore necessary to prototype, monitor, and predict the behavior of new CDN services/components in a controlled simulated environment, before and after their final release. As agent-oriented methods, particularly agent-based modeling and simulation (ABMS), have demonstrated high effectiveness for the analysis and design of modern complex distributed systems in a wide range of different application domains, this paper proposes an effective approach based on the agent-oriented ELDAMeth methodology for the design and analysis of CDNs. Specifically, an agent-based extensible CDN framework provides high-level programming abstractions and tools that fully support the simulation of different CDN architectures and mechanisms and allow for automatic evaluation of three main CDN performance indices: average user perceived latency, cache hit ratio and utility. The framework currently makes it available agent models for conventional, cooperative and clustered CDN architectures that have been simulated according to a reference simulation scenario. Results clearly show that clustered architectures have the potential to outperform conventional and cooperative architectures on which real CDNs are currently based and that the proposed agent-based framework is effective in modeling and analyzing CDNs.

1. Introduction

Content delivery networks (CDNs) are being extensively used in the Internet as the most effective solution for improving the performance of content delivery by means of coordinated content replication (Dilley et al., 2002; Peng, 2003; Pallis and Vakali, 2006; Buyya et al., 2008; Fortino and Mastroianni, 2009; Choi et al., 2011; Passarella, 2012). In particular, a CDN is an overlay network which manages a geographically distributed set of surrogate servers, located at the network edge, that archive copies of identical content. Users' content requests can be fulfilled by optimal surrogate servers instead of origin servers storing the original contents and located at the network core. As demonstrated in the CDN worldwide adoption (Nygren et al., 2010), the exploitation of CDNs actually leads to a significant reduction of latency times in accessing the requested content as well as to notable traffic alleviation of the network core. Nevertheless, the combined effects of some key factors such as the high decentralization of the CDN components over the best effort Internet, the mutable patterns of the content requests issued by the users, the specific redirection and coordination mechanisms among the CDN components, as well as the specific content request handling and content management policies of the surrogate servers, make CDNs artificial complex systems. Therefore, the design and analysis of CDNs become very complex tasks (Fortino et al., 2011) and require suitable methodologies and tools (Fortino and Mastroianni, 2008) able to model CDNs and analyze their complexity in terms of the aforementioned key factors.

To date, although several approaches have been proposed to model and analyze CDNs, the majority of them lack of flexibility and customizability as they address only specific CDN aspects, and usually are monolithic, scarcely reusable and extensible. Among the currently available approaches, it is possible to identify two main research efforts devoted to the generalization of the CDN modeling and analysis, CDNsim (Stamos et al., 2010), based on simulation, and MetaCDN (Broberg et al., 2009), based on a real overlay platform.

CDNsim (Stamos et al., 2010) enables users to evaluate and validate new policies and services under a realistic simulated CDN infrastructure. CDNsim is a public-source, modular and open-architecture parallel discrete-event trace-driven CDN simulation

* Corresponding author. Tel.: +39 0984 494063; fax: +39 0984 494713.
E-mail addresses: g.fortino@unical.it (G. Fortino), mvaccaro@si.deis.unical.it (W. Russo), w.russo@unical.it (M. Vaccaro).

1084-8045/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2012.11.005

system that is based on OMNeT++ (Varga and Hornig, 2008) simulation environment and the INET framework, an extension of OMNeT++ providing network protocols like TCP/IP; in particular, it exploits OMNeT++ only for the basic networking operations and for discrete-event scheduling. The request routing, content distribution, and management, as well as all the CDN characteristics, are simulated by CDNsim itself. The main features of CDNsim can be grouped in *CDN framework setup*, *content distribution and management*, and *request management*. The *CDN framework setup* includes CDN organization in surrogate servers and network components, servers categorized in origin and surrogate servers, relationships among clients, surrogate servers and origin servers, interaction mechanisms for communications among network elements based on TCP and among surrogate servers based on HTTP/FTP, and content service/types including static content, streaming media and services. The *content distribution and management* encompasses surrogate server placement, content selection and delivery, content outsourcing, and cache organization. The *request management* basically entails request-routing mechanisms based on DNS-based redirection. Although, CDNsim can be considered the most complete and full-fledged simulation environment for CDNs developed so far, it does not support clustering of surrogates (Fortino et al., 2009c), strategic to improve CDN performances, but only peer-to-peer cooperation among neighbor surrogates.

Differently from CDSsim, MetaCDN (Broberg et al., 2009) is a real system that leverages several existing "storage clouds", creating an integrated overlay network that provides a low cost, high performance content delivery network for content creators and providers. MetaCDN can place content onto one or many storage providers based on the quality of service, coverage and budget preferences of content creators/providers. This goal is fulfilled through a flexible connector-based architecture providing a uniform method/API for harnessing multiple storage clouds. Specifically, MetaCDN provides a single namespace to cover all supported providers, making it simple to integrate into origin sites, and handles load balancing for end-users. It offers content creators an easy way to harness the power of multiple cloud storage providers via a web portal. MetaCDN can also be used to evaluate performances of decentralized content delivery such as redirector overhead from specific areas, response time and throughput. Currently MetaCDN supports neither clustering of surrogate servers nor cooperation among surrogate servers. Although, the implementation of new policies and services atop MetaCDN is possible, this is not straightforward and needs skillful development efforts.

The analysis and design of modern complex distributed systems as CDN infrastructures require powerful and flexible methods, tools and techniques, which are also based on bottom-up approaches and incorporate the use of simulation to support the typical phases of a software engineering process (Hoffer et al., 2002). In this context, a suitable support can be represented by agent-based models of systems, i.e. system models which are built through a bottom-up approach in terms of proactive and reactive autonomous agents that dynamically interact and cooperate with each other (Garro and Russo, 2010). An agent-based model of a system can then be simulated to observe emergent macro-level phenomena hard to catch with other analysis techniques, and can be used to validate and evaluate different design choices at architectural and behavioral levels (Jennings, 2001). Moreover, the agent-based model of the system exploited during the design phase can also be used as a starting point for an agent-based system implementation (Cossentino et al., 2008; Fortino and Russo, 2012).

Agent-based modeling and simulation (ABMS) is emerged as an effective method to build agent-based models of systems and analyze them through simulation. ABMS is mainly applied to the analysis of biological, human and economics/financial complex systems. Several tools are currently available, such as Repast (North et al., 2006), Repast Symphony (North et al., 2007), Netlogo (Wilensky and Rand, in press; Niazi and Hussain, 2009), Swarm (Minar et al., 1996), ASCAPE (Inchiosa and Parker, 2002), and a few methodologies (Garro and Russo, 2010; Niazi et al., 2009) based on such tools have been also introduced. They are general-purpose tools and methodologies that fully support modeling and simulation of complex (adaptive) systems by providing flexibility, reusability, and extendibility. However, they are too general-purpose to effectively address the domain-specific issues raised by the CDNs and their related context. Such issues could be more effectively tackled by a more specific ABMS methodology that provides powerful abstractions for modeling artificial complex distributed systems and efficient mechanisms for their simulation.

In this paper, ELDAMeth (Fortino and Russo, 2012), an agent-oriented methodology for simulation-based prototyping of distributed agent systems, is used with the aim of defining an agent-based framework for the design and analysis of CDNs. ELDAMeth is particularly suitable for modeling and analyzing artificial complex distributed systems as it is characterized by: effective abstractions for modeling proactive components and their interactions, a multi-layered discrete event simulation framework for networked agent systems, a full-fledged development process seamlessly supporting the modeling and simulation phases. In particular, the defined framework can be exploited for analyzing already existing CDN solutions as well as for designing and evaluating new ones. Currently, it provides agent-based models for five CDN architectures: *Conventional*, *Cooperative*, *Master/Slave*, *Multicast-based*, and *Peer-to-Peer*. In *conventional* architectures, when a user request is redirected to a surrogate server, which is not able to fulfill it by providing the requested content, such surrogate server fetches the requested content from the origin server and delivers it to the requesting user. *Master/slave*, *multicast-based*, and *peer-to-peer* architectures (or *clustered* architectures) rely on a clustering schema (Fortino et al., 2009c), whereas the *cooperative* architecture depends on non-coordinated cooperation among neighbor surrogate servers. In particular, the *clustered* architectures differ from the *conventional* architecture as surrogate servers are grouped into clusters of neighbor surrogates which can cooperate. Specifically, a surrogate that is not able to provide the requested content does not directly ask the origin server for it as in the *conventional* architecture but it first checks for a surrogate of the same cluster having the content so as to forward the unfulfilled user request to it. Differently, in the *cooperative* architecture the surrogate asks its neighbors for the requested content and, upon content attainment, replies to the requesting user. Finally, both in *clustered* and *cooperative* architectures, if the surrogate is not able to find the content in its cluster or among its neighbors, it contacts the origin server as in the *conventional* architecture.

The defined agent models of the CDN architectures are obtained according to the following integrated phases of ELDAMeth:

- the *modeling* phase, which is driven by a modeling language based on Statecharts (Fortino et al., 2004, 2010b) and supported by a CASE tool (*ELDATool*) which automatically translates Statecharts-based visual specifications into platform-independent code;
- the *simulation* phase, which is based on an agent-oriented and event-driven simulation framework (*ELDASim*) (Fortino and Russo, 2012; Fortino et al., 2005c) that executes the code produced in the *modeling* phase upon a simulated distributed platform of agent servers.

An in-depth performance evaluation of the agent-based models representing the five CDN architectures has been carried out with respect to the most important CDN performance indices: average