An Architecture for Virtual Organization (VO)-Based Effective Peering of Content Delivery Networks

Al-Mukaddim Khan Pathan, James Broberg, Kris Bubendorfer*, Kyong Hoon Kim, Rajkumar Buyya

Grid Computing and **D**istributed **S**ystems (GRIDS) Lab
Department of Computer Science
and Software Engineering
The University of Melbourne, Australia
{apathan, brobergj, jysh, raj}@csse.unimelb.edu.au

*School of Mathematics Statistics
and Computer Science
Victoria University of Wellington
Wellington 6140, New Zealand.
kris@mcs.vuw.ac.nz

ABSTRACT

The proprietary nature of existing Content Delivery Networks (CDNs) means they are closed and do not naturally cooperate, resulting in "islands" of CDNs. Finding ways for distinct CDNs to coordinate and cooperate with other CDNs is necessary to achieve better overall service, as perceived by end-users, at lower cost. In this paper, we present an architecture to support peering arrangements among CDN providers, based on a Virtual Organization (VO) model. Our approach promotes peering among providers, reduces expenditure, while upholding user perceived performance. This is achieved through proper policy management of negotiated Service Level Agreements (SLAs) among peers. In addition, scalability and resource sharing among CDNs is improved through effective peering, thus evolving past the current landscape where "islands" of CDNs exist. We also show analytically that significant performance improvement can be achieved through the peering of CDNs.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.2 [**Computer-Communication Networks**]: Network Protocols; C.2.4 [**Computer-Communication Networks**]: Distributed Systems; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Distributed Systems*; H.3.5 [**Information Storage and Retrieval**]: Online Information service—*Web-based Services*

General Terms

Management, Measurement, Performance, Design, Economics.

Keywords

Architecture, Content Delivery Networks, Virtual Organization, Peering

1. INTRODUCTION

Content Delivery Networks (CDNs) provide services that improve network performance by maximizing available bandwidth, improving accessibility and maintaining correctness through content replication. Thus, they offer fast and reliable applications and services by distributing content to *edge* servers located close to end-users [2].

UPGRADE-CN'07, June 26, 2007, Monterey, California, USA.

Existing commercial CDNs are proprietary in nature. They are owned and operated by individual companies. Each of them has created its own closed delivery network, which is expensive to setup and maintain. Running a global CDN is even more costly, requiring an enormous amount of capital and labor. In addition, content providers typically subscribe to one CDN and thus can not use the resources of multiple CDNs at the same time.

Furthermore, commercial CDNs make specific commitments to their customers by signing Service Level Agreements (SLAs) [6]. An SLA is a contract between the service provider and the customer to describe provider's commitment and to specify penalties if those commitments are not met. So, if a particular CDN is unable to provide quality of service to the end-user requests, it may result in SLA violation and end up costing the provider.

One approach for reducing expenses and avoiding adverse business impact is to establish peering arrangements among CDN providers [13]. Such peering arrangements are appealing. However, the challenges in adopting such an arrangement include designing a system that virtualizes multiple providers and offloads end-user requests from the primary provider to peers based on cost, performance and load. In particular we identify the following key issues:

When to peer? The circumstances under which a peering arrangement should be triggered. The initiating condition must consider expected and unexpected load increases.

How to peer? The strategy taken to form a peering arrangement between multiple CDNs. Such a strategy must specify the interactions among entities and allow for divergent policies among peering CDNs.

Whom to peer with? The decision making mechanism used for choosing CDNs to peer with. It includes predicting performance of the peers, working around issues of separate administration and limited information sharing among peering CDNs.

How to manage and enforce policies? How policies are managed according to the negotiated SLAs. It includes deploying necessary policies and administering them in an effective way.

In this paper, we present a novel architecture of a Virtual Organization (VO) [10] based model for forming peering CDNs. In our architecture, a CDN serves end-user requests as long as the load can be handled by itself. If the load exceeds its capacity, the excess end-user requests are offloaded to the Web servers of the peers. The VO-based peering system endeavors to cut expenses, improving locality while also upholding user perceived performance. The main contributions of this paper are:

- an architecture for an open and decentralized system that supports the effective peering of CDNs within a scalable VO-based model, and
- a policy-based framework for SLA negotiation among peering CDNs to ensure that requests are effectively served, meeting user QoS requirements.
- a preliminary analytical model based on the fundamentals of queuing theory to demonstrate the performance gain through the proposed VO-based peering of CDNs.

The rest of the paper is structured as follows: Section 2 presents the related work. Section 3 presents our architecture for peering CDNs with a broad view of the VO creation steps and VO formation scenarios. It also provides a description on the architectural components and features. Section 4 describes the policy management in peering CDNs environment, with a focus on negotiated SLAs and defined policy levels. Section 5 outlines analytical model for peering CDNs. Finally, Section 6 concludes the paper with a brief summary of expected contributions and future directions.

2. RELATED WORK

Peering of content delivery networks is gaining popularity among researchers of the scientific community. Several projects are being conducted for finding ways to peer CDNs for better overall performance. In this section, we outline some of these efforts.

The internet draft by IETF proposes a Content Distribution Internetworking (CDI) Model [11], which allows CDNs to have a means of affiliating their delivery and distribution infrastructure with other CDNs who have content to distribute. The CDI Internet draft assume a federation of CDNs but it is not clear how this federation is built and by which relationships it is characterized. A protocol architecture for CDI is presented in [12]. The main drawback of this protocol is – being a point-to-point protocol, if one end-point is down the connection remains interrupted until that end-point is restored.

A peering system for content delivery workloads in a federated, multi-provider infrastructure has been presented in [13], but the peering strategy, resource provisioning and performance guarantees among partnering CDNs are unexplored in this work. CDN brokering [14] allows one CDN to intelligently redirect end-users dynamically to other CDNs in that domain. The drawback is that, the routing mechanism used is proprietary in nature and might not be suitable for a generic CDI architecture. From a user-side perspective, Cooperative Networking (CoopNet) [8] addresses the flash crowd problem through the cooperation of end-hosts. The main problem of the user-side mechanisms is that they are not transparent to end-users, which are likely to restrict their widespread deployment.

Other systems such as Coral [17], CoDeeN [18], Globule [15], and DotSlash [16] address the issue of collaborative content delivery. However, they do not virtualize multiple providers for cooperative management and delivery of content in a peering environment.

3. VO-BASED PEERING CDNS

A CDN is expected to provide the necessary distributed computing and network infrastructure to ensure SLAs are met with its customers. In order to meet such SLAs and to manage its resources properly, it could be necessary to cooperate with other CDNs. In our architecture, cooperation among the peering CDNs is achieved through a VO. Formation of a VO is initiated by a CDN, which realizes that it will not be able to meet its SLAs with the customers. The initiator is called a *primary* CDN; while other CDNs who share their resources in a VO are called *peering* CDNs. Users interact transparently with the VO by requesting content from Web servers of the primary CDN. A content request may initiate further VO activities that the end-users are unaware of (e.g. inter-CDN request-routing, content replication and delivery in a peering arrangement). Thus, the participating entities act as a single conceptual unit in the execution of common goal(s).

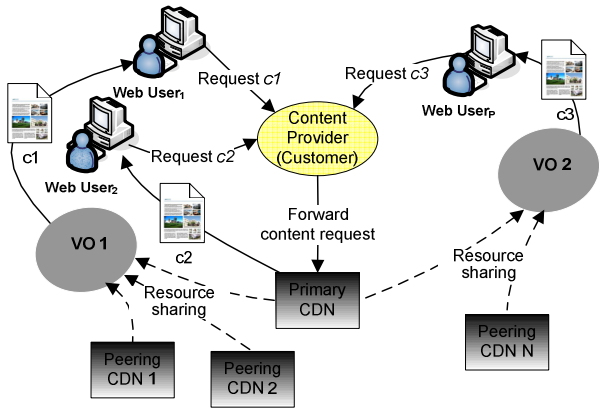

Figure 1: A formed VO

A VO is composed of *explicit* members who are the primary and any peering CDNs who cooperate for resource sharing, and *implicit* members who are content providers and end-users. Implicit members are transparent to a VO but they share the benefit from it. Figure 1 shows the example of VO-based peering CDNs. End-user requests for content are forwarded to the primary CDN which is holding the content from the content provider. The requested content is served either directly by the primary CDN or by any peering CDNs within a VO. Let us consider content c1 and c3 in Figure 1. Since c1 and c3 reside in the Web servers within VO1 and VO2 respectively, requests of c1 and c3 are served accordingly from VO1 and VO2. In case of content c2, the primary CDN directly delivers the requested content.

Table 1: List of commonly used terms

Terminology	Description
Web server (WS)	A Container of content
Mediator	A policy-driven entity, authoritative for policy negotiation and management
Service registry (SR)	Discovers and stores resource and policy information in local domain
Peering Agent (PA)	A resource discovery module in the peering CDNs environment
Policy repository (PR)	A storage of Web server, mediator and VO policies
P_{WS}	A set of Web server-specific rules for content storage and management
P_{M}	A set of mediator-specific rules for interaction and negotiation
P_{VO}	A set of VO-specific rules for creation and growth of the VO

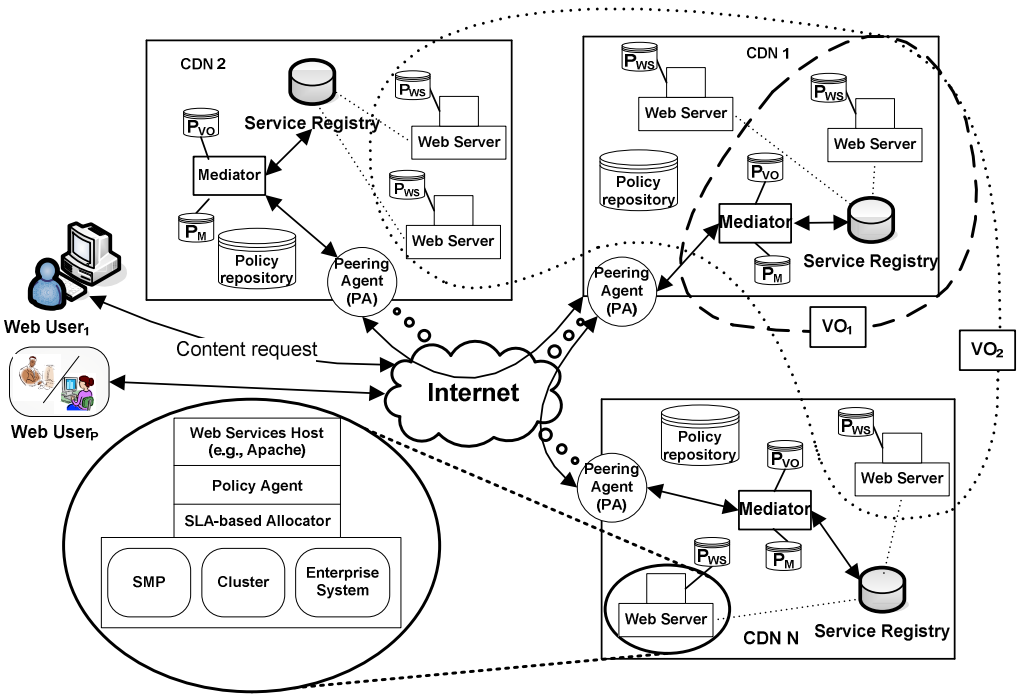

Figure 2: Architecture of a system to assist the creation of peering CDNs

3.1 System architecture

The architecture of our VO-based peering CDNs is shown in Figure 2. The terminologies used to describe the system architecture are listed in Table 1. In the VO-based peering CDNs model, a CDN endeavors to balance its service requirements against the high costs of deploying customer- dedicated, over-provisioned resources. Thus, to cut expenses and avoid the potential peak load threat of violating SLAs with the customers, CDNs will be able to leverage computing and network infrastructure from other CDNs through peering. The negotiation among CDNs for resource peering allows a *peering* CDN to agree to allocate the required amount of its local resources (Web servers, bandwidth etc.) to provide content and services on behalf of the *primary* CDN.

A peering arrangement among CDNs that provisioning and sharing of computing resources must also provide settlement and exchange of generated revenue. The primary CDN ultimately controls the resources it has acquired – which are delegated rights for the peering CDNs' physical resources. The physical resources could consist of resources from multiple peering CDNs distributed over different geographical locations. The primary CDN determines what proportion of the Web traffic (i.e. end-user requests) is redirected to the Web servers of the peering CDNs, which content is replicated there, how the replication decisions are taken, and which replication policies are being used.

In our architecture, *Web Servers (WS)* within a CDN are the actual holders of content. Each Web server has its own policies, defined as a set of server-specific rules, P_{WS}, for the storage and management of content. The *Service Registry (SR)* helps in discovering local resources within a CDN by providing resource and access related information. The *Peering Agent (PA)*, *Mediator* and *Policy Repository (PR)* collectively act as a "gateway" for a given CDN, and all three assist in creation of a new VO. The PA of a CDN acts the role of a resource discovery module. It acts as the first point of contact for other CDNs when they are initiating a peering agreement, and a conduit through which a CDN can itself discover potentially useful resources available from other CDNs. The mediator is responsible for negotiation among CDNs and management of operations within a VO. The mediator has its own policies (defined as a set of mediator-specific rules, P_M) and also manages the policies (defined as a set of VO-specific rules, P_{VO}) necessary for negotiation and creation of Virtual Organization(s). The PR virtualizes all of the individual policies from within the VO, including P_{WS}, P_M, and P_{VO}, and will ultimately include policies from peering CDNs.

3.2 Lifecycle of a VO

A VO may vary in terms of purpose, scope, size, and duration. Hence, VOs are of two types: *short-term on-demand VOs* and *long-term VOs* with established SLAs. A short-term VO is formed for limited duration, based on current user request patterns to prevent the generation of *hotspots* [9]. Such a peering arrangement should be automated to react within a tight time frame – as it is unlikely that a human directed negotiation would occur quickly enough to satisfy the evolved niche. A short-term VO is formed on-demand and the policy for such VO formation is established dynamically to handle the situation, one such negotiation mechanism is briefly described in section 3.4. Short-term VOs are phased out when the workload returns to normal. On the other hand, a long-term VO is formed for an event which will be known in advance. In a long-term VO, CDNs collaborate for longer period of time and such a VO remains for the duration of the event. In this situation we would expect negotiation to include a human-directed agent to ensure that any resulting decisions comply with participating companies' strategic goals.

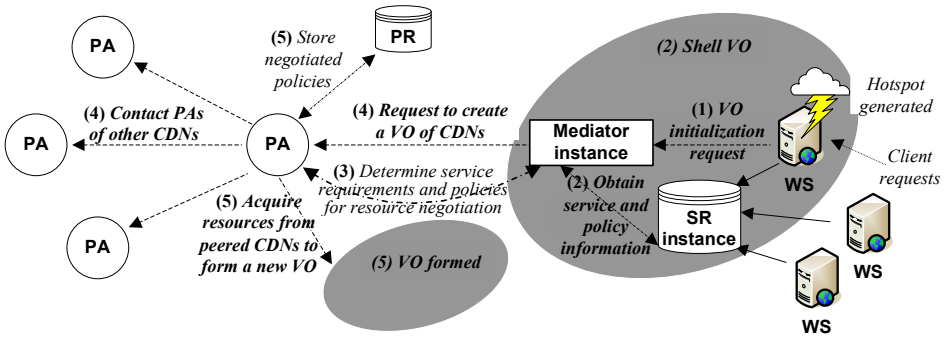

Figure 3: The formation of a Virtual Organization (VO)

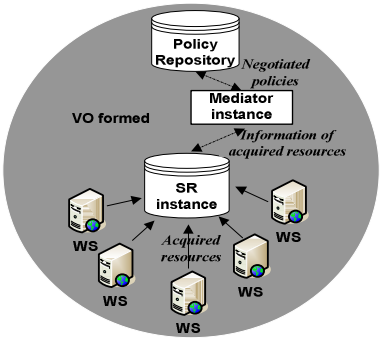

Figure 4: A formed VO

Relevant scenarios for short and long-term VO creation have been illustrated in [1].

Figure 3 illustrates the VO creation steps, while Figure 4 shows a VO after it is formed. The followings are typical steps for a VO creation:

Step 1. *A (primary) CDN provider realizes that it cannot handle a part of the workload on its Web server(s). A VO initialization request is sent to the mediator.*

Step 2. *The primary CDN constructs a shell VO, with a mediator instance, a service registry instance, and a policy registry. The mediator instance obtains the resource and access information from the SR, whilst SLAs and other policies from the PR.*

Step 3. *The shell VO represents the potential for peering of resources. Hence, it needs to be expanded to include additional resources from other CDNs. The mediator instance on the primary CDN's behalf generates its service requirements based on the current circumstance and SLA requirements of its customer(s).*

Step 4*. The mediator instance passes the service requirements to the local Peering Agent (PA). If there are any preexisting peering arrangements (for a long term scenario) then these will be returned at this point. Otherwise, it carries out short term negotiations with the PA identified peering targets (Section 3.4).*

Step 5 *When the primary CDN acquires sufficient resources from its peers to meet its SLA with the customer, the new VO becomes operational. If no CDN is interested in such peering, VO creation through re-negotiation is resumed from Step 3 with reconsidered service requirements.*

An existing VO may need to either disband or re-arrange itself if any of the following conditions hold: (a) the circumstances under which the VO was formed no longer hold; (b) peering is no longer beneficial for the participating CDNs; (c) an existing VO needs to be expanded further in order to deal with additional load; or (d) participating CDNs are not meeting their agreed upon contributions.

3.3 Architectural components

In this section, we provide the description of the architectural components along with their responsibilities:

Web server – Web Servers are responsible for storing content and delivering them reliably. We separate a server's structure into two layers: *overlay* and *core*. In the overlay layer, a server comprises a Web-service host (for example, Apache or Tomcat), a policy agent, and a SLA-allocator. The Web services host ensures the delivery of content to end-users based on the negotiated policies. The policy agent is responsible (in conjunction with the mediator) for determining which resources can be delegated and under what conditions (policies) delegation is permitted. The SLA-allocator performs the provisioning and reservation of Web server's resources (e.g. CPU, bandwidth, storage etc.) to satisfy both local and delegated SLAs, and ensures that the terms of the SLAs are enforced. The Web server's core consists of high performance computing systems such as symmetric multiprocessors, cluster systems, or other enterprise systems (such as desktop grids). The Web servers' underlying algorithms perform on-demand caching, content selection, and routing between servers. This requires each Web server to express its own policies for storage and management of content.

Mediator – The (resource) mediator is a policy-driven entity in a VO-based peering CDNs environment. The raison d'etre for an instance of the mediator within a VO is to ensure that the participating entities are able to adapt to changing circumstances (agility) and are able to achieve their objectives in a dynamic and uncertain environment (resilience). Once a VO is established, the mediator controls what portion of the Web traffic (i.e. end-user requests) is redirected to the Web servers of the peering CDNs, which content is replicated there, how the replication decision is taken, and which replication policies are being used. When performing automated peering the mediator will also direct any decision making during peering negotiations (during VO

creation), policy management, and scheduling. A mediator holds the initial policies for VO creation and obtains additional composite polices as a result of successful peering negotiations. A mediator works in conjunction with its local Peering Agent (PA) to discover external resources and to negotiate with other CDNs. An example of a mediator led negotiation is given in Section 3.4.

Service Registry (SR) – The SR encapsulates the resource and service information for each CDN. It helps in discovering local resources through enabling the Web servers of CDN providers to register and publish their resource, service and policy details. In the face of traffic surges, the service registry is accessed by the mediator in shell VO creation to supply any necessary local resource information. When a shell VO is grown to form a new VO, an instance of the service registry is created that encapsulates all local and delegated external CDN resources.

Policy Repository (PR) – The PR virtualizes all of the policies within the VO. It includes the Web server-specific policies, mediator policies, VO creation policies along with any policies for resources delegated to the VO as a result of a peering arrangement. These policies form a set of rules to administer, manage, and control access to VO resources. They provide a way to manage the components in the face of complex technologies.

Peering Agent (PA) – The PA is a CDN specific entity that exists prior to a VO creation. It is independent of any VO. It acts as a policy-driven resource discovery module for VO creation and is used as a conduit by the mediator to exchange policy and resource information with other CDNs. It is used by a primary CDN to discover the peering CDNs' (external) resources, as well as to let them know about the local policies and service requirements prior to commencement of the actual negotiation by the mediator.

3.4 Short-term resource negotiation

In order to respond to hotspots that may result in a CDN failing to meet its QoS obligations, we propose that time-critical agreements for a short-term VO should be automatically negotiated. We expect any such agreements to hold for a limited duration and only involve an artificially restricted set of CDN resources. Even so, there are serious issues involving such agreements including trust, i.e. who governs the allocation decisions and are they trustworthy; and the potential commercial sensitivity of information about the current state and costs of a CDN. Divulging commercially sensitive information (e.g. resources, access and policies) as a basis for negotiating a peering arrangement would be, in general, commercially unacceptable. Even with the limitations placed on resources that can be automatically delegated, there must be checks and balances to ensure that any delegation is made properly. Otherwise, it would also be unlikely that a CDN would agree to have an external party (e.g. mediator of the primary CDN) make allocations of their resources and bind them to negotiated SLAs.

One solution to these problems is to utilize a cryptographically secure auction [22], which hides both the valuations that CDNs place on their resources and who is participating in the auction. In this case all the mediators would between themselves act as a secure distributed auctioneer, in which the cryptographic protocol itself guarantees a trustworthy outcome of the auction. These auctions have been shown to be tractable in practice and are therefore an ideal basis for automation of peering agreements. A negotiation would start with the VO being formed and the mediator determining the shortfall in resources. The mediator then issues its local PA a call for bids and within this includes the SLAs that it requires. The PA distributes the request to other PAs of the peers and each of them then passes the request to its CDN mediator. All mediators that wish to bid then register with the requesting mediator and a subset of the mediators (acting as the *distributed auctioneers*) are selected via a cut and shuffle [23]. Note that the requesting mediator does not act as an auctioneer. The auction is then held securely and only the final resource allocations are revealed.

An auctioneer starts an auction not for selling an item (i.e. allocation), but for buying it. Buyers (peering CDNs) bid with the price they are willing to sell the allocation of their Web servers. One bidder can not see the bid of other bidders. An auctioneer gathers bids from the bidders and selects the lowest bidding agent(s) as the winner and the winner is paid second-lowest bidding price. In other words, a reverse Vickrey auction is used. As mentioned earlier, we assume that an auction is held using a cryptographically secure auction [22] protocol to hide all auction related sensitive information. Through this approach, a mendacious behavior from a provider is restricted. Thus, over-provisioning of resources by harnessing data through VO membership, or modifying and falsifying of content by some rogue CDN providers is not allowed. Here below, we also summarize the steps for the auction to be held within a VO:

Step 1. *The mediator of the primary CDN (buyer) realizes the need of additional resources to replicate content. It internally determines the maximum payable amount (expressed by Payoff Value). The mediator then issues its local PA a call for bids and within this includes the SLAs that it requires (Auction Policy).*

Step 2. *The PA distributes the request to other PAs of the peers and each of them then passes the request to its CDN mediator.*

Step 3. *All mediators that wish to bid then register with the requesting mediator and a subset of the mediators are chosen to act as distributed auctioneers.*

Step 4. *All bidders (peering CDNs) use a Bidding Function to determine the bidding amount.*

Step 5. *An auctioneer collects bids and selects the lowest bidding buyer(s) as winner and a winner is paid by the amount of second-lowest bid.*

Step 6. *An auction takes place successfully when winners are chosen according to the Auction policy of the primary CDN. At this point, it can be assumed that the primary CDN has acquired sufficient resources from its peers to meet its SLA with customers. Therefore, the primary CDN replicates its content to the winning CDNs' Web servers. If no winner is selected through the auction, re-negotiation through auction takes place, starting from Step 1.*

3.5 Architectural features

The operation of a CDN is driven by semi-autonomous logic that ensures content is served reliably through content replication, request-routing and redirection whilst maintaining constant awareness of the health (e.g. load information) of participants.

Request assignment and *redirection* can be performed in a CDN at multiple levels – at the DNS, at the gateways to local clusters and also (redirection) between servers in a cluster [19][20]. Commercial CDNs predominantly rely on DNS level end-user

assignment combined with a rudimentary request assignment policy (such as weighted round robin, or least-loaded-first) which updates the DNS records to point to the most appropriate replica server [7]. In the proposed framework, end-users can be assigned via DNS (by the peering agents of participating CDNs updating their DNS records regularly) and also via redirection at the CDN gateway when appropriate.

Content replication occurs from origin servers to other servers within a CDN. Existing CDN providers (e.g. Akamai, Mirror Image) use a non-cooperative pull-based approach, where requests are directed (via DNS) to their closest replica server [7]. If the file requested is not held there, the replica server pulls the content from the origin server. Co-operative push-based techniques have been proposed that pushes content onto participating mirror servers using a greedy-global heuristic algorithm [21]. In this approach, requests are directed to the closest mirror server, or if there is no suitable mirror nearby, it is directed to the origin server. In the context of our peering framework, this replication extends to participating servers from other CDNs in a given VO, subject to the available resources it contributes to the VO. This is defined by the policies (P_{VO}) agreed upon at VO creation time.

Load information can be measured and disseminated within individual CDNs and amongst other CDNs. A load index can provide a measure of utilization of a single resource on a computer system. Alternatively, it can be a combined measure of multiple resources like CPU load, memory utilization, disk paging and active processes. Such load information needs to be disseminated amongst all participating CDNs in a timely and efficient manner to maximize its utility. Such indices will also be crucial to identify situations where forming a VO-based peering is appropriate (e.g. when servers or entire CDNs are overloaded) or when CDNs resources are underutilized and could be offered to other CDN providers. Within the VO-based framework, we anticipate a hierarchical approach, where current bandwidth and resource usage of web servers in a CDN is reported to the CDN gateway (i.e. mediator, PA and policy repository as a single conceptual entity) in a periodic or threshold-based manner. The gateways of participating CDNs then communicate aggregated load information describing the load of their constituent servers.

4. SLA NEGOTIATION AND POLICY MANAGEMENT

When CDNs peer according to a VO-based model, the participants sign SLAs with different performance objectives. Once the SLAs have been agreed upon, the participants in a VO work in order to satisfy the negotiated SLAs. The SLA components include:

- *Description of service requirements*, a specification of the resource and service requirements of the primary CDN. This description includes the storage requirements, the required rate of transfer, the primary CDN's preference to gain resources at a particular region, and the expected duration of receiving service.
- *Administration for VO activities*, which specifies the role of the mediator as an authoritative entity in the VO.
- *Renegotiation for problem resolution*, which illustrates the steps to be undertaken in face of any problem in providing necessary services.
- *Consequences of SLA violation*, which outlines the possible results of SLA violation in which service expectations are not met. The consequences of SLA violation may range from imposing penalty on the participants through reimbursement of part of the revenues lost due to the loss of service, to termination of peering relationship, and to disbanding and/or rearranging the participants to form a new VO.
- *SLA bypassing conditions*, which details the conditions under which the SLAs are not applicable. Such situations include the damage of physical resources due to natural disaster, theft etc.

4.1 Policy management to support SLAs

The proper operations of a VO-based peering CDNs architecture seek for the consistent performance and availability of a large number of widely distributed system entities, specified in a Service Level Agreement (SLA). A policy-based framework can simplify the complexities involved in the operation and management of a large content distribution network [5]. Within our VO-based peering CDNs architecture we apply the standard policy framework defined by the IETF/DMTF [4]. We also define three levels of policies, namely – Web server policy, mediator policy and VO policy. These three policy levels are detailed in Section 4.2.

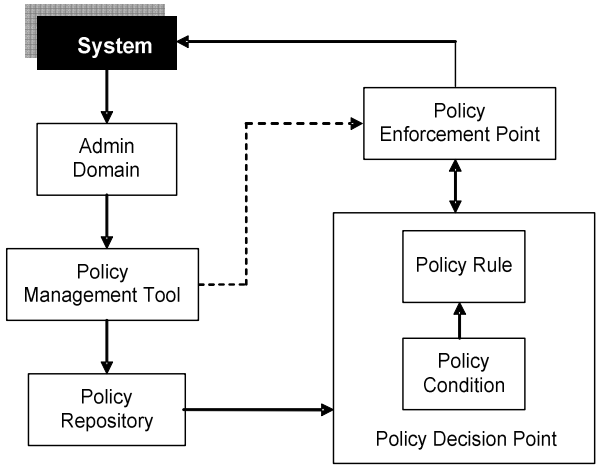

Figure 5: Basic policy framework

We define a policy as a descriptive statement that allows an entity to properly administer, manage, and execute its activities. Policies in the context of peering CDNs are statements that specify how the Web servers should deal with different types of traffic, what type of content can be moved out to a CDN node, what resources can be shared between the VO participants, what measures are to be taken to ensure quality of service based on negotiated SLAs, and what type of programs/data must be executed at the origin servers. Thus these policies endeavor to provide a way to manage multiple entities deploying complex technologies within the architecture of the VO-based peering CDNs.

We illustrate the standard policy framework in Figure 5. In the standard policy framework, the *admin domain* refers to an entity which administers, manage and control access to resources within the *system* boundary. An administrator uses the *policy management tools* to define the policies to be enforced in the system. The *policy enforcement points (PEPs)* are logical entities within the system boundary, which are responsible for taking

Table 2: Policy mapping

Policy framework Component	Peering CDNs Component	Specified policies	Description
System	*Peering CDNs*	All policies in the system	The distributed computing and network infrastructure for peering CDNs
Admin domain	*Formed VO*	Negotiated VO policies	An administrative entity for resource management and access control
Policy management tool	*Administrator dependent*	–	An administrator dependent tool to generate policies
Policy repository	*Policy repository*	Web server, VO and mediator policies	Storage of policies in the system
Policy Enforcement Points (PEPs)	*Web Services host, Policy Agent, SLA-based allocator*	Web server policies	A logical entity which ensures proper enforcement of policies
PDPs	*Mediator*	Mediator policies, VO policies	An authoritative entity for retrieving policies from the repository

action to enforce the defined policies. The *policy repository* stores polices generated by the administrators using the policy management tools. The *policy decision point (PDP)* is responsible for retrieving policies from the policy repository, for interpreting them (based on *policy condition*), and for deciding on which set of policies are to be enforced (i.e. *policy rules*) by the PEPs.

The model (and the corresponding entities) for peering CDNs can be mapped to the basic policy framework. We show this mapping in Table 2. The policy repository (PR) from Figure 2 virtualizes the Web server, mediator and VO policies. These policies are generated by the policy management tool used by the VO administrator. The distribution network and the Web server components (i.e. Web Services host, Policy Agent, SLA-based Allocator) are the instances of PEPs, which enforce the peering CDN policies stored in the repository. The mediator is the instance of the PDPs, which specifies the set of policies to be negotiated at the time of collaborating with other CDNs, and passes them to the peering agent at the time of negotiation. The policy management tool is administrator dependent and it is not shown in Figure 2.

4.2 Policy levels

In this section we delineate different levels of policy definitions that are present within the domain of peering CDNs. We propose following a general implementation of policies which can be specified according to their granularity level. Being influenced by [5], we also argue for storing the specification of the policy rules (for all three levels of policies) in the policy repository for maintaining interoperability. We also initiate simplified management by storing the definition of policies in the policy repository, and by defining abstractions that provide a machine dependent specification of policies. We propose that the policy specification contained in the policy repository should be defined in terms of the technology to which a policy would apply, rather than in terms of the configuration parameters of any specific resource.

Here below, we describe the distinct policy levels:

Web server policies specify,

- how a server performs consistent content caching;
- how the policy agent module operates based on negotiated SLAs; and
- how the SLA-based allocator module ensures provisioning of resources to satisfy negotiated SLAs.

Mediator policies specify,

- how the mediator interacts with the PA to pass information on service requirements;
- how the mediator takes over the administration of delegated resources once they are acquired;
- how the mediator effectively manages VO activities to cope with changing circumstances; and
- how the mediator coordinates with other entities to assist in resource allocation.

VO policies include,

- the policies necessary for initiating VO creation;
- the policies need to be administered in face of SLA violation by VO participants; and
- the policies to dynamically disband or rearrange a VO.

5. ANALYTICAL MODELING

In this section, we develop a simple analytical model based on the fundamentals of queuing theory to demonstrate the performance gain through the peering of CDNs.

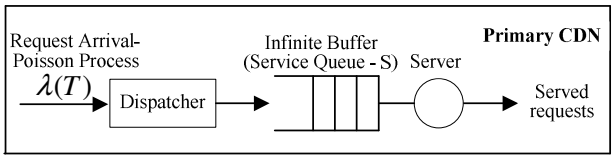

Figure 6: Primary CDN modeled as an M/G/1 queue

Let us model a CDN as an M/G/1 queue as shown in Figure 6. Table 3 shows the parameters and expressions that are used in the analytical modeling. An M/G/1 queue consists of a FIFO buffer with requests arriving randomly according to a Poisson process at rate λ and a processor, called a server, which retrieves requests from the queue for servicing. We assume that the total processing of the Web servers of a CDN is accumulated through the server and the service time is a general distribution. User requests are serviced at a first-come-first-serve (FCFS) order. We use the term 'task' as a generalization of a content request arrival for service. We denote the processing requirements of an arrival as 'task size.' Here, we will use the terms task and request arrival interchangeably. Such a content request can be a client requesting

Table 3: Parameter and expressions for the analytical model

Parameter	Expression
Mean arrival rate	$\lambda = (1/T)$ (requests/second)
Mean arrival time	T
Mean service rate	μ
Load	$\rho = (\lambda/\mu)$
P. D. F of service distribution	$f(x) = \frac{\alpha k^{\alpha}}{1-(k/p)^{\alpha}} x^{-\alpha-1}$, $k \le x \le p$
Task size variation	α
Smallest possible task size	k
Largest possible task size	p
Expected waiting time	$E(W)$
Mean service time	$E(X)$

an individual file or object, a Web page (containing multiple objects), the results of execution of a script (e.g. CGI, PHP) or any digital content.

We abstract all the request streams coming to the Web servers of the primary CDN as a single request stream. Client requests arrive at a conceptual entity, called *dispatcher*, following a Poisson process with the mean arrival rate λ. All requests in its queue are served on a FCFS basis with mean service rate μ. It has been observed that the workloads in Internet are heavy-tailed in nature [24][25], characterized by the function, $\Pr\{X > x\} \sim x^{-\alpha}$, where $0 \le \alpha \le 2$. In a CDN, clients request for content of varying sizes (ranging from small to large). Based on size of the content requested, the processing requirements (i.e. task size) also vary. Thus, the task size on a given CDN's service capacity follows a Bounded Pareto distribution. The probability density function for the Bounded Pareto $B(k, p, \alpha)$ is

$$f(x) = \frac{\alpha k^{\alpha}}{1-(k/p)^{\alpha}} x^{-\alpha-1},$$

where α represents the task size variation, k is the smallest possible task size, and p is the largest possible task $(k \le x \le p)$. By varying the value of α, we can observe distributions that exhibit moderate variability $(\alpha \approx 2)$ to high variability $(\alpha \approx 1)$.

We start with the derivation of the expectation of waiting time $E(W)$. W is the time a user has to wait for service. $E(N_q)$ is the number of waiting customers and $E(X)$ is the mean service time. By Little's law, the mean queue length $E(N_q)$ can be expressed in terms of the waiting time. Therefore, $E(N_q) = \lambda E(W)$ and load on the server, $\rho = \lambda E(X)$. Let $E(X^j)$ be the *j*-th moment of the service distribution of the *tasks*. We have,

$$E(X^j) = \begin{cases} \frac{\alpha p^j ((k/p)^{\alpha} - (k/p)^j)}{(j-\alpha)(1-(k/p)^{\alpha})} & \text{if } j \neq \alpha \\ \frac{\alpha k^{\alpha} \ln(p/k)}{(1-(k/p)^{\alpha})} & \text{if } j = \alpha \end{cases}$$

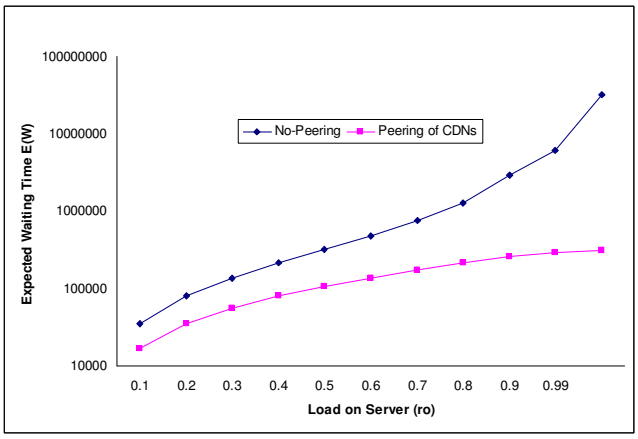

Figure 7: Effectiveness of peering among CDNs

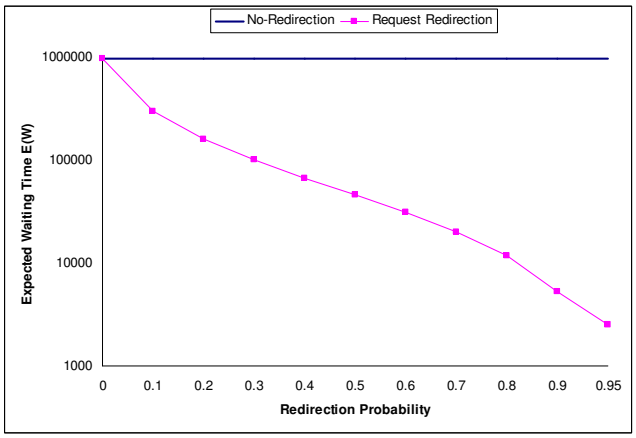

Figure 8: Impact of request-redirection on expected waiting time

Hence, using P-K formula, we obtain the expected waiting time in the primary CDN's queue, $E(W) = \lambda E(X^2)/2(1-\rho)$. Now we want to measure the expected waiting time with respect to varying server load and task sizes.

5.1 Performance gain through peering

Now, let us assume that we have *N* peering CDNs in the system. All the participants share their resources to deal with flash crowds. In face of sudden surge in demand, the load on the primary CDN becomes, $\overline{\rho} = \rho(1 - P_{redirect})$ and the redirected load is distributed equally among the peering CDNs. Figure 7 shows the effectiveness of peering among CDNs in terms of expected waiting time for a two CDNs case. In Figure 7, we can see a reasonable improvement in expected waiting time through peering. Without peering when the system load approaches to 1.0, the user perceived response time for service by the primary CDN tends to infinity. In this case, the primary CDN peers with other CDNs for coordinated and cooperative delivery of content. Hence, as the primary CDN becomes overloaded, it redirects some of its request to the peers. In this way, even in the face of high demand, user perceived performance remains satisfactory. Figure 8 shows the impact of request-redirection on the expected waiting time with high system load $(\rho = 0.95)$ with task variability, $\alpha = 1.5$. In Figure 8, we show that as the redirection probability

increases, approaching to 1, expected waiting time on the primary CDN decreases substantially.

5.2 Three CDN peering scenario

In this section, we examine the performance gain through peering in three CDN peering arrangement, as illustrated in Figure 9. Each of the participating CDNs is modeled as an M/G/1 queue. As shown in Figure 9, when the primary CDN is unable to serve all the incoming content requests, it redirects a fraction of the request to the peering CDN, which in return serve those requests. Note that, the redirected requests are equally distributed between the peering CDNs. Under this assumption, we can measure the expected waiting time for end-user requests for each of the three participating entities in the peering arrangement.

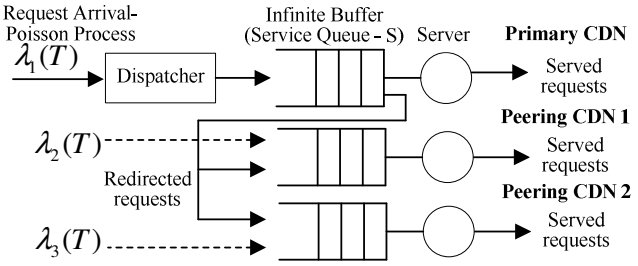

Figure 9: A peering scenario with three CDNs

Figure 10 shows the impact of redirecting the requests on expecting waiting time for the three CDNs peering scenario and Table 4 lists the notations used for this scenario. Initially, the primary CDN and peer 1 have moderate system load ($\rho = 0.5$), whereas peer 2 is less loaded ($\rho = 0.3$). As the load on the primary CDN increases (approaching to 1.0), it offloads content requests to the peers, with redirection probability, $P_{redirect} = 0.5$. For this scenario we assume that content requests are coming as different request streams to the participating CDNs. We also assume that each of the participating entities has the same service rate. The initial and new load on CDN i is measured by, $\rho_i = \lambda_i E(X_i)$ and $\overline{\rho}_i = \overline{\lambda}_i E(X_i)$ respectively. λ_i is the initial arrival rate, whereas, $\overline{\lambda}_i$ is the new arrival rate at CDN i. We calculate $\overline{\lambda}_i$ as,

$$\overline{\lambda}_i = \begin{cases} (1 - P_{redirect})\lambda_i & \text{if } i = 1 \\ \lambda_i + \lambda_1(\frac{P_{redirect}}{N}) & \text{if } i > 1 \end{cases}$$

We measure the initial and new expected waiting time of CDN i as,

$$E(W_i) = \frac{\lambda_i E(X_i^2)}{2(1-\rho_i)} \text{ and}$$

$$E(\overline{W}_i) = \frac{\overline{\lambda}_i E(X_i^2)}{2(1-\overline{\rho}_i)}, \text{ respectively.}$$

From Figure 10, we find that, the expected waiting time of the primary CDN decreases as the requests are redirected to the peering CDNs. Peering CDNs, on the other hand, show an increase in expected waiting time as it receives extra requests from the primary CDN.

Table 4: List of notations in three CDNs peering scenario

Notation	Description
N	Number of CDNs, $N \in \{1,2,...,N\}$
$E(X_i^j)$	j-th moment of CDN i's service distribution
ρ_i	Initial load on CDN i, $\rho_i = \lambda_i E(X_i)$
$\overline{\rho}_i$	New load on CDN i, $\overline{\rho}_i = \overline{\lambda}_i E(X_i)$
λ_i	Initial arrival rate at CDN i
$\overline{\lambda}_i$	New arrival rate at CDN i, $\overline{\lambda}_i = \begin{cases} (1 - P_{redirect})\lambda_i & \text{if } i = 1 \\ \lambda_i + \lambda_1(\frac{P_{redirect}}{N}) & \text{if } i > 1 \end{cases}$
$E(W_i)$	Expected waiting time (initial) at CDN i, $E(W_i) = \frac{\lambda_i E(X_i^2)}{2(1-\rho_i)}$
$E(\overline{W}_i)$	Expected waiting time (new) at CDN i, $E(\overline{W}_i) = \frac{\overline{\lambda}_i E(X_i^2)}{2(1-\overline{\rho}_i)}$
$P_{redirect}$	Probability to redirect content requests from CDN i

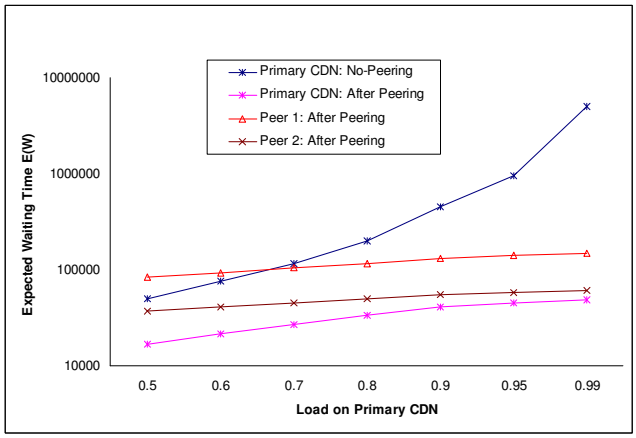

Figure 10: Three CDNs scenario - performance gain through peering

6. CONCLUSION AND FUTURE WORK

In this paper, we present an open and scalable system to assist the creation of open content delivery networks. In our architecture, when the load on the primary CDN exceeds its capacity, it peers with other CDNs, and the excess end-user requests are offloaded to the Web servers of the peering CDNs. Our contribution lies in designing an architecture for the VO-based peering approach that endeavors to reduce setup and maintenance cost of network infrastructures, while also respecting end-user performance requirements through proper policy management for negotiated SLAs. It also promotes extended scalability and resource sharing with other CDNs through cooperation and coordination. We also constructed a preliminary analytical model for peering CDNs demonstrating the performance gain through peering. Such results highlight the utility of peering among the existing CDNs. We

anticipate that, with more advanced modeling the proposed framework will motivate and direct our research in finding best practice techniques in measuring and disseminating load information, performing request assignment and redirection, and enabling content replication for CDNs participating in VO-based peering.

No prior work done in the content internetworking domain has considered VO-based peering among CDNs. Many of them make strong assumptions on the characteristics of applications without virtualzing multiple providers for cooperative management and delivery of content in a peering environment. Moreover, none of these systems have explored the issue of policy management.

Our future work includes using market models in this context in order to encourage resource sharing and peering among different CDNs at global level. This approach is inspired by the successful utilization of economic concepts in management of autonomous resources in global grids [3]. The use of economic concepts in this context would provide a solid basis for rational agents to decide whether to join in peering arrangements. The use of economic models may be the basis for a dynamic replication mechanism that makes replication decisions to utilize surrogates in areas which exhibit the potential to generate hotspots. Our initial work on this issue can be found in [1].

We expect that a VO policy driven model for forming CDNs will be a timely contribution to the ongoing content-networking trend. For more information, please visit the project Web site at www.gridbus.org/cdn.

REFERENCES

[1] Pathan, A. M. K. and Buyya, R. Economy-based content replication for peering CDNs. TCSC Doctoral Symposium, In *Proc. of the 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2007)*, Brazil, May, 2007.

[2] Buyya, R., Pathan, A. M. K., Broberg, J., and Tari, Z. A case for peering of content delivery networks. *IEEE Distributed Systems Online*, 7(10), USA, Oct. 2006.

[3] Buyya, R., Abrahamson, D., and Venugopal, S. The Grid economy. *Proc. of the IEEE*, 93(3), pp. 698-714, 2005.

[4] Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S, Huynh, A., Carlson, M., Perry, J., and Waldbusser, S. Terminology for policy based management. IETF RFC 3198, Nov. 2001.

[5] Verma, D. C., Calo, S., and Amiri, K. Policy-based management of content distribution networks. *IEEE Network*, pp. 34-39, March/April 2002.

[6] Bouman, J., Trienekens, J., and Zwan, M. Specification of service level agreements, clarifying concepts on the basis of practical research. In *Proc. of the Software Technology and Engineering Practice Conference*, pp. 169, 1999.

[7] Dilley, J., Maggs, B., Parikh, J., Prokop, H., Sitaraman R., and Weihl, B. Globally distributed content delivery. *IEEE Internet Computing*, pp. 50-58, Sept./Oct. 2002.

[8] Padmanabhan, V. N. and Sripanidkulchai, K. The case for cooperative networking. In *Proc. of International Peer-To-Peer Workshop (IPTPS02)*, 2002.

[9] Adler, S. The SlashDot Effect: An analysis of three Internet publications. *Linux Gazette*, Vol. 38, 1999.

[10] Norman, T. J., Preece, A., Chalmers, S., Jennings, N. R., Luck, M., Dang, V. D., Nguyen, T. D., Deora, V., Shao, J., Gray, W. A., and Fiddian, N. J. Agent-based formation of virtual organizations. *Knowledge-Based Systems*, 17 (2-4), pp. 103-111, 2004.

[11] Day, M., Cain, B., Tomlinson, G., and Rzewski, P. A model for content internetworking. IETF RFC 3466, Feb. 2003.

[12] Turrini, E. An architecture for content distribution internetworking. Technical Report UBLCS-2004-2, University of Bologna, Italy, March 2004.

[13] Amini, L., Shaikh, A., and Schulzrinne, H. Effective peering for multi-provider content delivery services. In *Proc. of 23rd Annual IEEE Conference on Computer Communicatitons(INFOCOM'04)*, pp. 850-861, 2004.

[14] Biliris, A., Cranor, C., Douglis, F., Rabinovich, M., Sibal, S., Spatscheck, O., and Sturm, W. CDN brokering. *Computer Communications*, 25(4), pp. 393-402, 2002.

[15] Pierre, G. and Steen, M. Globule: A platform for self-replicating Web documents. In *Proc. of the 6th International Conference on Protocols for Multimedia Systems (PROMS'01)*, Enschede, The Netherlands, pp. 1-11, 2001.

[16] Zhao, W. and Schulzrinne, H. DotSlash: A self-configuring and scalable rescue system for handling Web hotspots effectively. In *Proc. of the International Workshop on Web Caching and Content Distribution (WCW)*, Beijing, China, pp. 1-18, 2004.

[17] Freedman, M. J., Freudenthal, E., and Mazières, D. Democratizing content publication with Coral. In *Proc. of the 1st Symposium on Networked System Design and Implementation (NSDI'04)*, San Francisco, CA, pp. 239-252, 2004.

[18] Wang, L., Park, K. S., Pang, R., Pai, V. S., and Peterson, L. Reliability and security in the CoDeeN content distribution network. In *Proc. of the USENIX 2004 Annual Technical Conference*, 2004.

[19] Colajanni, M., Yu, P. S., and Dias, D. M. Analysis of task assignment policies in scalable distributed Web-server systems. *IEEE Transactions on Parallel and Distributed Systems*, 9(6), pp. 585-600, June 1998.

[20] Cardellini, V., Colajanni, M., and Yu, P. S. Request redirection algorithms for distributed Web systems. *IEEE Transactions on Parallel and Distributed Systems*, 14(4), pp. 355-368, April 2003.

[21] Cardellini, V., Colajanni, M., and Yu, P. S. Efficient state estimators for load control policies in scalable Web server clusters. In *Proc. of the 22nd Annual International Computer Software and Applications Conference*, 1998.

[22] Bubendorfer, K. and Thomson, W. Resource management using untrusted auctioneers in a Grid economy. In *Proc. of the 2nd IEEE International Conference on eScience and Grid Computing*, Dec. 2006.

[23] Ogston, E. and Vassiliadis, S. A peer-to-peer agent auction. In *Proc. of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2002.

[24] Crovella, M. E., and Bestavros, A. Self-similarity in World Wide Web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6), pp. 835-846, 1997.

[25] Crovella, M. E., Taqqu, M. S., and Bestavros, A. Heavy-Tailed Probability Distributions in the World Wide Web. *A Practical Guide To Heavy Tails*, Birkhauser Boston Inc., Cambridge, MA, USA, pp. 3-26, 1998.