Web Caching and Content Delivery
A. Bestavros and M. Rabinovich (Editors)

Synopsis

Multicast Injection for Application Network Deployment

Oscar Ardaiz [1], Felix Freitag [2], Leandro Navarro [3]

Computer Architecture Department, Polytechnic University Catalonia, Barcelona, Spain

Abstract

Introduction of new services on the Internet is a laborious, time-consuming task. Application networks, applications being serviced through multiple interconnected service nodes disseminated across the Internet, are as well costly to set up. This paper investigates dynamic deployment mechanisms to realize application network service introduction on the Internet.

1. Introduction

Application networks, such as content distribution networks, have been shown to improve service performance, availability and fault tolerance. Today application networks are provisioned either by individual resource providers replicating services uncoordinatedly or by centralized management stations activating services on every managed node. In the first case service providers are not able to control service quality, whereas the management station model is not scalable and does not consider shared third party resources. Introduction of application network services by dynamic deployment mechanisms aims at providing services with a predefined service level making a cost-effective use of shared resources.

2. Application Network Deployment

Application network deployment is defined as the process of setting up or creating an application network service. It includes all steps from service specification up to making it available to users. Deploying a service requires: obtain service specifications, map specifications to resources, discover resources, gather resources, configure resources, activate service, and provide a management interface.

We implement this functionality making use of an architecture composed of resource agents at resource providers' nodes and deployment managers at service providers' nodes. Resource agents are responsible for publishing resources, mediating between resources and service providers' deployment managers, configuring resources, activating

[1] E-mail: oardaiz@ac.upc.es
[2] E-mail: felix@ac.upc.es
[3] E-mail: leandro@ac.upc.es

services, and returning management interfaces. Deployment managers are responsible for obtaining service specifications, mapping specifications to resources, discovering resources, gathering resources, trading with resource agents on behalf of service providers, and managing overall deployment operation.

Service providers input the following service characteristics to create service spec: service type (static content, streaming content, servlet, etc.), service name, per surrogate storage, network coverage (AS or IP-subnet list), per region expected traffic, maximum distance between clients to surrogates (num hops, RTT, bandwidth, etc.), number of surrogates providing service, service duration and service start time.

Deployment mechanisms make service deployers discover and gather resources, plus activating services on appropriate nodes. Agents at those nodes activate services by configuring local surrogates, installing servlets, etc. Deployment managers obtain nodes location to configure redirection subsystems.

3. Multicast Injection

Dynamic deployment mechanisms have to discover and gather resources in a subset of all available nodes, instead of in every node, to be cost-effective. A first solution is the SNMP management station based method, in which a centralized entity monitors, chooses and configures resource agents. It is not scalable nor caters for contention at shared third party resources. Our approach is to use multicast based mechanisms to achieve scalability, adaptability, and simple agents. Multicast injection considers resource agents as active entities that can decide autonomously whether to accept or discard a service activation request based on local policies and information they announce to each other. Periodic refreshes and soft state ensures adaptability to demand and network variations.

Multicast injection deployment takes the following steps: (1) Deployer injects application service specifications in a global multicast channel. (2) Resource agents map those specifications to its resource capabilitities and availability, if they can provide service to some regions they allocate resource and announce their intention to start service in a service multicast channel. (3) After a short random interval, if they have not listened to other agents indicating willingness to provide service with better characteristics, they activate the service. (4) Periodically: They announce a service active plus resources available message. If they do not receive similar messages from enough nodes they conclude there are not enough resources and cancel service. If other nodes provide best service for equal regions they stop it.

We have performed simulations [1] to compare with centralized deployment how cost-effectively proposed mechanisms scale and adapt to demand and network variations.

4. Related Work

Xbone [3] dynamically deploys IP overlays with topology constraints by computing centralized allocations at overlay managers. Cluster reserves [2] perform centralized CPU-time resource allocations for geographically distributed Web servers on non-shared resource nodes. Our proposal aims at designing mechanisms to allow any entity to deploy application networks, or to provide resources for third party application activation.

References

[1] Ardaiz O., Freitag F., Navarro L., Multicast Injection for Application Network Deployment. UPC-DAC-2001-18.
[2] Aron M., Druschel P., Zwaenepoel W. Cluster reserves: A mechanism for resource management in cluster-based network servers. *Proc. ACM SIGMETRICS 2000*.
[3] Touch J., Dynamic Internet Overlay Deployment and Management Using the X-Bone. *Proc. ICNP 2000*.