Future Generation Computer Systems 28 (2012) 24–35

Contents lists available at SciVerse ScienceDirect

Future Generation Computer Systems

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

Overlay network resource allocation using a decentralized market-based approach[☆]

Jay Smith [a,b,*], Edwin K.P. Chong [b,d], Anthony A. Maciejewski [b], Howard Jay Siegel [b,c]

[a] DigitalGlobe, Longmont, CO 80503, USA

[b] Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523–1373, USA

[c] Department of Computer Science, Colorado State University, Fort Collins, CO 80523–1373, USA

[d] Department of Mathematics, Colorado State University, Fort Collins, CO 80523–1373, USA

ARTICLE INFO

Article history:
Received 18 March 2011
Received in revised form
10 June 2011
Accepted 16 July 2011
Available online 27 July 2011

Keywords:
Robustness
Dynamic resource management
Market-based resource allocation
Heterogeneous computing
Overlay network

ABSTRACT

We present a decentralized market-based approach to resource allocation in a heterogeneous overlay network. This resource allocation strategy dynamically assigns resources in an overlay network to requests for service based on current system utilization, thus enabling the system to accommodate fluctuating demand for its resources. Our approach is based on a mathematical model of this resource allocation environment that treats the allocation of system resources as a constrained optimization problem. From the solution to the dual of this optimization problem, we derive a simple decentralized algorithm that is extremely efficient. Our results show the near optimality of the proposed approach through extensive simulation of this overlay network environment. The simulation study utilizes components taken from a real-world middleware application environment and clearly demonstrates the practicality of the approach in a realistic setting.

1. Introduction

Recently, information technology systems have begun to rely heavily on the concept of Services Oriented Architecture (*SOA*). SOA is a means of leveraging existing applications as services within a distributed computing environment to develop new applications. A framework commonly used to integrate existing applications is known as the enterprise services bus (*ESB*) [1]. According to the Business Integration Journal [2], "The [ESB] supports the unifying integration infrastructure required for SOA and heterogeneous environments". By relying on an ESB to communicate with service providers, service requesters need not depend on the details of service provider implementations, e.g., the physical location of the service provider. Instead, service requesters depend on the abstract definition of the service that they are using and trust the ESB to forward their requests to an appropriate service provider. In our environment, we model the ESB as an overlay network that interconnects service providers and service requesters. Request forwarding is provided by a *broker* service that is deployed within the overlay network.

Because the service requester relies on the broker service and is not explicitly dependent on any single instance of a service provider, multiple service providers could be deployed within the ESB to provide additional capacity for a service that is in high demand. In this context, *capacity* is defined in terms of the number of service requests that can be processed per unit time. To take advantage of the added capacity provided by these additional service providers, the broker service deployed within the ESB serves as a proxy for communication between the service requesters and the service providers. That is, the broker service allocates incoming service requests to the collection of service providers for processing.

Although the ESB is commonly deployed as a single centralized platform, the scalability and reliability of the ESB can be greatly improved through replication of the platform. A common approach to increasing the reliability of a system is to duplicate that system many times across many hardware deployments, a technique often referred to as *replication* [3,4]. An important aspect of an ESB implementation is that it can be decentralized both to increase its reliability and to ensure its scalability [2,4].

Successfully replicating brokers within an ESB requires maintaining network transparency [3]; i.e., the user of the system

☆ This research was supported by the NSF under Grants CNS-0615170, CNS-0905399, and ECCS-0700559, and by the Colorado State University George T. Abell Endowment.

* Corresponding author at: DigitalGlobe, Longmont, CO 80503, USA. Tel.: +1 720 839 5378.

E-mail addresses: jtsmith@digitalglobe.com (J. Smith), echong@colostate.edu (E.K.P. Chong), aam@colostate.edu (A.A. Maciejewski), hj@colostate.edu (H.J. Siegel).

0167-739X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2011.07.002

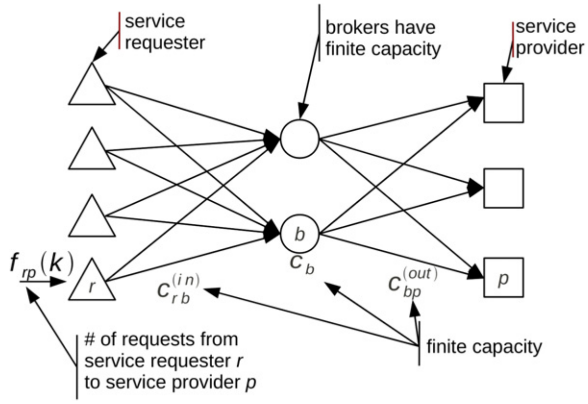

Fig. 1. An example system where four service requesters are utilizing two brokers to communicate with three service providers. Service requesters are shown as triangles, brokers as circles, and service providers as squares. Each input link to a broker from a service requester has a finite capacity, denoted $c_{rb}^{(\text{in})}$. Likewise, each output link from a broker to a service provider has a finite capacity denoted $c_{bp}^{(\text{out})}$. Finally, each broker has a finite capacity for servicing requests, denoted c_b.

should be shielded from the existence of any redundant components used to provide the ESB or its attached services. To the user, the broker service should appear as a single highly available system that always has sufficient capacity to route service requests. Achieving this kind of transparency in service delivery requires a mechanism for allowing the broker service to adapt dynamically to the changes in system load. That is, transparently utilizing replica service providers deployed within the ESB requires that the broker provide a mechanism for routing requests to their physical destination that can be decentralized. In this work, we focus on the allocation of resources to the effective transmission of service requests to service providers.

In this work, we investigate the application of a decentralized market-based approach to resource management within a heterogeneous deployment of an ESB platform. Service requesters send tasks to service providers using an overlay network provided by the ESB. The decision of how to allocate broker capacity to service requests is made in a decentralized manner based on a quantification of current resource demand relative to current system capacity. Individual service requesters select transmission rates through Brokers that maximize their individual benefit, while the brokers adjust "prices" for network links and broker computing capacity to reflect current demand for shared resources. Thus, price setting enables service requester decision making by enabling shared resources to communicate a simple quantification of current system congestion to requesters.

Fig. 1 presents a graphical model of a simple overlay network provided by an ESB. Service requesters are depicted in the figure as triangles, service providers as squares, and brokers as circles. Each service requester is connected to a collection of brokers that "service" requests by dispatching them to a service provider capable of completing the request. The number of requests produced by each service requester may vary with time according to some unknown process. The capacity of the brokers to service incoming requests may differ from one component to another, i.e., the collection of brokers are assumed heterogeneous in their performance [5–7]. Finally, each broker is connected to a collection of service providers by a finite capacity link.

Our mechanism for resource management can be thought of as a market-based approach where market demand for shared resources helps the system to set prices for those resources. Some market-based approaches rely on an auction to create a market where prices are set by the highest bidder [8–12]. In this environment, the service requesters, that would be bidding for service in an auction based mechanism, are part of the resource management mechanism and are cooperating to maximize the utility of the overall system. Because the service requesters are not adversarial participants in the system the complexity of an auction based mechanism is not required.

Our approach utilizes price setting based on duality theory [13]. In duality theory, selected constraints are directly accounted for in the optimization criterion as a penalty. This approach is analogous to that used in congestion control on the Internet [14], in ad-hoc sensor networks [15–17], and server provisioning within large scale distributed clusters [18]. In our system model, price variables are introduced to model market demand for shared resources, where prices provide a simple quantification of demand relative to supply. For example, as the number of requests through a broker increases, the broker raises its "price". Conversely, if the number of requests through a broker decreases, its quoted price also decreases. In addition to measuring the demand for the component itself, the broker also is responsible for stating the demand for the links from that broker to all of the service providers with which the broker can communicate. The procedure used to calculate optimal prices for resources is presented in Section 3.

In this study, we are considering an environment where all brokers are controlled by a single organization. As such, the brokers are assumed to be cooperating to provide an efficient system. We assume that all brokers use the common pricing mechanism presented in Section 3 and as such do not bias their pricing by over or under stating demand for shared resources. Individual service requesters directly utilize current pricing information provided by the brokers to make local resource allocation decisions about how best to assign their volume of requests within the overlay network, given current network utilization. Obviously, this requires that each service requester is logically connected to every broker within the system. Using all of these components in concert, we will show that our decentralized mechanism results in provably optimal resource allocations.

In the replicated broker environment, we assume that each service provider offers a unique service to the overall system and that each service provider always has sufficient capacity to service all incoming requests. In a real system, a service provider may be implemented using a collection of finite-capacity replica providers, where the replicas combine to provide a more reliable scalable service implementation. Within each of these collections, we treat the allocation of finite-capacity service provider resources as a separate resource allocation problem. In Fig. 2, we have re-drawn the distributed environment to show how the collection can be used to provide such a service to the system. When treated in isolation, the allocation of service provider capacity within each collection is simpler than our original distributed broker problem because there is only one class of shared resource to manage.

To demonstrate resource allocation in this simpler environment, we apply our approach to a related distributed web hosting environment [19]. In a distributed web hosting environment, the hosted web site is replicated to multiple web servers to increase the apparent reliability and performance of the web site. Incoming user-driven HTTP requests for data are routed to the web servers for processing by a collection of independent service requesters. By indirecting web server access through service requesters, we can allocate incoming user requests using a decentralized approach that is similar to that of the ESB environment. The overlay network topology of this environment includes only service requesters and service providers, where a service provider is defined to be a web server. This two-layer overlay network can also be used to model the allocation of requests by brokers to replicated service providers as in our previous example. That is, in this simpler model, the brokers act as service requesters to a collection of replicated service providers.

The contributions of this work include a new mathematical model of resource management in an overlay network that treats the allocation of shared resources to service requests as