Chapter 8
Overview of Service Oriented Architecture for Resource Management in P2P Systems

Farag Azzedin
King Fahd University of Petroleum & Minerals, Saudi Arabia

Mohamed Eltoweissy
Virginia Tech - Advanced Research, USA

Salman Ahmad Khwaja
King Fahd University of Petroleum & Minerals, Saudi Arabia

ABSTRACT
The P2P computing is one of the technologies that is having a significant impact on the way Internet-scale systems are built. It is well established for applications such as file sharing and parallel distributed computation. The well establishment for P2P systems is built on the notion of sharing. One of the objectives of resource sharing is to increase some measure of the overall work done by the collection of resources. The authors present a service oriented architecture for sharing and managing resources in P2P systems. As P2P services have shown expedient way of interaction between peers utilizing different services, our architecture will extend the high utilization of P2P networks with service orientation for handling more services with better management and easy extendibility to further capitalize on the success of P2P networks.

INTRODUCTION
Resource-sharing is one of the most popular attributes of Peer-to-Peer (P2P) systems, where peers do not need to set up and operate servers in order to provide resources used by others. Different P2P networks come to existence for different types of resources and purposes. For example, Gnutella is used for file-sharing and SETI@home is used for distributed computing. In order to efficiently utilize the sharing of resources, a Resource Management System (RMS) must be put in place. A resource management system manages the resources that comprise the computing environment (Traversat, 2002).
Managing resources in a P2P environment is a challenging task because of the lack of centralized control and the intermittent nature of peers. Resource management in contemporary P2P system usually depends on the network overlay. Napster uses a client-server approach in handling resource discovery and uses a P2P approach in transferring the resource. On the other hand, Gnutella uses flooding to discover resources since there is no special peer to handle resource discovery. Furthermore, a FastTrack network dynamically differentiates peers into regular nodes and Supernodes. Supernodes handle most of the resource management tasks.

These approaches to RMS in a P2P setting have certain limitations. The use of centralized entity such as in Napster becomes a single point of failure that can make the whole system non-functional. On the other extreme, flooding messages in Gnutella can take considerable bandwidth which results in scalability problems as the number of nodes increases (Ritter, 2001). Lv, Ratnasamy & Shenker, (2002) studied the scalability of Gnutella, where a design was proposed to: (a) restrict the query flow into each node so they do not become overloaded and (b) dynamically evolve the overlay topology so that queries flow towards the nodes that have sufficient capacity to handle them.

The support of only one resource type limits the utilization as well as the widespread of P2P applications such as Gnutella and KaZaA. Furthermore, DHT-based systems assume that the peers have uniform capabilities. This assumption, however, is difficult to be accepted in real dynamic systems (Hu, Wang & Zhu, 2005). Nodes have varying availability rate, bandwidth, storage, and computing power.

We hypothesize that the limitations of P2P systems can be addressed by a Service Oriented Architecture (SOA) approach. Peers can provide and use different services without having to know the internal workings of these services. Each peer does not need to implement all services as it can use services of other peers to perform its tasks. This is when re-use comes in to the game. A peer now is able to quickly build up another solution that will reuse a set of these services while also providing additional services that the application may require. Such services are woven in to a sort of ecosystem of software. In this scenario, they cooperate as a means of achieving some sort of business objective, while possibly participating in some other ecosystems that offer similar service capability for a possibly different end solution.

The basic idea of SOA is not entirely new. The idea is that it would be nice to create complex systems from simple separate parts. SOA can be defined as an architectural style of building reliable distributed systems that deliver functionality as services, with the additional emphasis on loose coupling between interacting services as defined by Treadwell (2005). Furthermore, SOAs stay independent of the internal workings of the services. As such, it needs to provide certain standards defining the services, their inputs and outputs. Anyone can provide the service as long as the standards for the service are met. A peer can choose the best service for its needs as the peers are loosely-coupled. This can encourage service providers to improve their quality of service in order to attract more business.

Binding SOA with a P2P network seems one of the most logical solutions, as P2P is already providing services to the millions of the networked users. As such, adding SOA on P2P systems will introduce more flexibility and more diversity. The flexibility is provided by allowing different types of service and resource providers to participate in the network. These different service and resource providers give rise to diversity of services available to all peers on the network. The recent interest in SOA was spurred by the emergence of web services (Krajicek & Kuba, 2006). Current incarnations of web services and P2P systems suffer from distinct and, in many cases complimentary drawbacks. While web services are strong on standards, they show possible weaknesses on issues such as scalability and resilience to node
Related Content

A Two-Level Fuzzy Value-Based Replica Replacement Algorithm in Data Grids

Effective Integration of Reliable Routing Mechanism and Energy Efficient Node Placement Technique for Low Power IoT Networks
www.igi-global.com/article/effective-integration-of-reliable-routing-mechanism-and-energy-efficient-node-placement-technique-for-low-power-iot-networks/188790?camid=4v1a

Service-Oriented Symbolic Computations
www.igi-global.com/chapter/service-oriented-symbolic-computations/40839?camid=4v1a

Architectural Elements of Resource Sharing Networks
www.igi-global.com/chapter/architectural-elements-resource-sharing-networks/36423?camid=4v1a