IMPLEMENTATION OF SESSION-STATE-SERVICE-A FRAMEWORK FOR P2P NETWORK

S. JAYAVARDHANARAO¹, B. VENKATA RAMANA²

¹Department of IT, Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh, India
Email: jayavardhanarao.s@gmail.com
²Assoc. professor, Department of IT, Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh, India.

Abstract
— Peer-to-Peer systems have become, in a short period of time, one of the fastest growing and most popular applications. The decentralized and distributed nature of P2P systems leads to living aside the client-server model. In P2P model each node takes both the roles of client and server. As a client, it can query and download its wanted data files from other nodes and as a server, it can provide data files to other nodes. Two main objectives in p2p systems are data location and search for interesting data. In order to present these topics, this paper will focus on the issue of peer-to-peer session state service for networks, which will provide the security and good service.

Index Terms — P2P, session state service, node, client-server.

I. INTRODUCTION

Peer-to-peer networking is the utilization of the relatively powerful computers (personal computers) that exist at the edge of the Internet for more than just client-based computing tasks. The modern personal computer (PC) has a very fast processor, vast memory, and a large hard disk, none of which are being fully utilized when performing common computing tasks such as e-mail and Web browsing. The modern PC can easily act as both a client and server (a peer) for many types of applications[1,2].

The typical computing model for many applications is a client/server model. A server computer typically has vast resources and responds to requests for resources and data from client computers. Client computers initiate requests for resources or data from server computers[3]. A good example of the client/server model of computing is Web browsing. Web servers on the Internet are typically high-end dedicated server computers with very fast processors (or multiple processors) and huge hard disk arrays[4,5]. The Web server stores all of the content associated with a Web site (HTML files, graphics, audio and video files, etc.) and listens for incoming requests to view the information on a particular Web page. When a page is requested, the Web server sends the page and its associated files to the requesting client. Peer-to-peer networking has the following advantages over client/server networking:

• Content and resources can be shared from both the center and the edge of the network. In client/server networking, content and resources are typically shared from only the center of the network.
• A network of peers is easily scaled and more reliable than a single server. A single server is subject to a single point of failure or can be a bottleneck in times of high network utilization.
• A network of peers can share its processor, consolidating computing resources for distributed computing tasks, rather than relying on a single computer, such as a supercomputer.

II. LITERATURE SURVEY

Peer-to-peer research draws upon numerous distributed systems disciplines. Networking researchers will recognize familiar issues of naming, routing and congestion control. P2P designs need to address routing and security issues across network region boundaries (Sollins 2003). Networking research has traditionally been host-centric. The web’s Universal Resource Identifiers are naturally tied to specific hosts, making object mobility a challenge (Walfish, Balakrishnan et al.2004)

P2P work is data-centric (Shenker 2003). P2P systems for dynamic object location and routing have borrowed heavily from the distributed systems corpus. Some have used replication, erasure codes and Byzantine agreement (Rhea, Wells et al. 2001). Others have used epidemics for durable peer group communication (Gupta, Birman et al. 2003). Similarly, P2P research is set to benefit from database research (Gribble, Halevy et al. 2003).

Database researchers will recognize the need to reapply Codd’s principle of physical data independence, that is, to decouple data indexes from the applications that use the data (Hellerstein 2003). It was the invention of appropriate indexing mechanisms and query optimizations that enabled data independence. Database indexes like B+ trees have an analog in P2P’s distributed hash tables.
(DHTs). Wide-area, P2P query optimization is a ripe, but challenging, area for innovation.

III. SYSTEM ARCHITECTURE

The idea behind the peer to peer state server is simple -- let state servers on a network securely communicate and pass session state data amongst each other as needed, as shown below. This design improves scalability because web servers can share multiple state servers, eliminating a single point of failure. Furthermore, if a load balancer erroneously or intentionally redirects a user to a different web server attached to a separate state server, the user’s session state will be requested from the state server that served the user’s previous request. Security is also improved as peers can be configured to encrypt session data while sharing session state. Data transfers between the web server and the state server remain unencrypted but eavesdropping attacks can be eliminated by keeping web servers and linked state servers in trusted networks or on the same computer. The peer to peer state server is fully backward compatible with the Microsoft provided state server and comes with all the benefits mentioned earlier shown in Fig 1.

IV. WORKING PROCEDURE:

The Peer to Peer State Server works exactly as illustrated above, except when the state server doesn’t have the requested session state, in which case it requests the session state from the network before responding, as illustrated Fig 2.

1. Web server requests to read, write or perform some other action on a session state.

2. State server 1 does not have the requested session state from connected peers and then waits for the session state to be transferred.

3. State servers 2 and 3 do not have the requested session state, so they forward the message to their connected peers.

4. The message is continuously forwarded until it arrives at a peer that has the requested session state, which is in this network is server 7.

5. State server 7 connects directly to server 1 and transfers the requested session state.

6. State server 1 responds to the web server request.
If the requested session state is not transferred within a set time period, the state server assumes the session state does not exist on the network and proceeds to process the web server request without the session state. The GetTransferMessage class represents the message that is broadcast on the network when a node is requesting a session. Peers maintain connection between themselves principally to forward this message. Session state transfers occur out-of-band of the peer network.

V. IMPLEMENTATION

Various programming techniques are used to implement different aspects of the state server. Some of the notable ones are highlighted below.

V. IMPLEMENTATION

A. PLATFORM

The state server is written in C# 2.0 but targets the .NET 3.5 framework so as to take advantage of the ReaderWriterLockSlim class. If the NET20 symbol is defined, the server uses the slower ReaderWriterLock class instead and is able to target the .NET 2.0 framework. This tool will be extremely useful as I spec out my implementation of the state service. For instance, if I want to see how the state service reacts if there is no Exclusive header, I’ll simply modify the TransformData method to detect and remove the header as shown in the fig 3.

Fig 3 Results
VI. CONCLUSIONS

The peer to peer state server is fully backward compatible with the Microsoft provided state server and can transparently replace it. Peer state servers can transfer sessions to each other, improving the reliability of session state dependent web applications. Peer state servers also act as a security layer that protects session data on the network.

References

[4]. Ranjan, Rajiv; Harwood, Aaron; Buyya, Rajkumar (1 December 2006), A Study on Peer-to-Peer Based Discovery of Grid Resource Information
[5]. Lua, Eng Keong; Crowcroft, Jon; Pias, Marcelo; Sharma, Ravi; Lim, Steven (2005). "A survey and comparison of peer-to-peer overlay network schemes".