WebPeer: A P2P-based System for Publishing and Discovering Web Services

Ruixuan Li

Internet and Distributed Computing Laboratory
College of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan 430074, P.R.China
rxli@hust.edu.cn
Outline

- Background of P2P Research
- A Glance of our Current Projects
- WebPeer: A Web Services Oriented Peer-to-peer System
- Experiments & Implementation
- Summary & Ongoing Work
Outline

- Background of P2P Research
- A Glance of our Current Projects
- WebPeer: A Web Services Oriented Peer-to-peer System
- Experiments & Implementation
- Summary & Ongoing Work
Society Development

First phase: Before B.C. 2000

Hominid society: cooperative equal commutative
Structured society:
- Hierarchical
- Ranking
- Centralized

Second phase:
- Before
- Now
- ……
- The Day After Tomorrow

The Day After Tomorrow

IDC Lab, HUST
Service Architecture Development

Client/Server Architecture

Cooperative Architecture

The Arch. After Those

Peer-to-Peer Architecture!
Predictions

☐ From Forbes (Feb., 2005)

- (Internet) Applications based on peer-to-peer topologies will be the mainstream.

☐ From Brainpower of U.S (Jan., 2005)

- Self-Aware Peer-to-Peer Systems will develop resilient, scalable sensor/computation networks with decentralized control.
Distributed Computing Economics

(Views of Jim Gray)

- **An equivalent price** for following computing items:
  - one database access
  - 10 bytes of internet traffic
  - 100,000 instructions
  - 10 bytes of disk storage
  - a megabyte of disk bandwidth

- The break-even point is **10,000 instructions per byte**

- This serves a basis how we do **cost-effective Internet-based computing**, such as peer-to-peer computing
What is Peer-to-Peer?

- A model of communication where every node in the network acts alike.
- As opposed to the Client-Server model, where one node provides services and other nodes use the services.
- In P2P network, every node is both client (consumer) & server (producer).
Why P2P?

- Inherent scalability
- Abundant resources
- No central point of failure
- No guarantee about QoS
Building P2P Topology

Hybrid Centralized P2Ps
- Napster
- Central Metadata
- Single Failure Point
- Low Scalability

Pure Decentralized P2Ps
- Gnutella, Freenet
- No Central Point
- Good Scalability
- Flooding-based Search
- Hard Management

Partially Decentralized P2Ps
- KazaA, Morpheus
- Structured
- Good Scalability
P2P Applications

- File Sharing
- Science Computing
- Collaboration
- E-Commerce
- Others
Key Issues

- Topology Maintenance
- Searching Scheme (Routing Protocol)
- Data Dissemination Scheme
- Buffer Management
- Security and Reputation
Our Experiences on P2P

- **WebPeer**
  - A Web Services Oriented P2P System
  - [http://idc.hust.edu.cn/webpeer/](http://idc.hust.edu.cn/webpeer/)

- **CoEdit**
  - A P2P Based Collaborative Editing System
  - [http://idc.hust.edu.cn/coedit/](http://idc.hust.edu.cn/coedit/)

- **TrustedPeer**
  - A Secure and Dynamic Trusted P2P System
Next …

- Background of P2P Research
- A Glance of our Current Projects
- WebPeer: A Web Services Oriented Peer-to-peer System
- Experiments & Implementation
- Summary & Ongoing Work
CoEdit: Overview

- **Collaborative Editing** is a form of Editing which involves a group editing process.

- **Scenario**: Cartographers, designers and drawers fulfill drawing objectives, through sharing resources, context and group interaction.

![Diagram of CoEdit](image)

- Monitoring Activities
- Joining Activities
- Evaluating Activities

Collaborative Workplace
- Discussion
- Collaborative Editing
- Collaborative Drawing etc

Internet/Intranet

- Shared Knowledge
- Construction

Cartographer

Drawer

Drawer

Designer
CoEdit: Issues

- **Session Management**: How do distributed users create, destroy, join and leave collaborative sessions?

- **Concurrency Control**: How do we ensure that concurrent users do not enter inconsistent commands, or merge concurrent commands entered by different users?

- **Undo/Redo**: What are the semantics of undo/redo in a collaborative session?

- **Awareness**: How are users made aware of “out of band” activities of their collaborator?

- **Access Control**: How do we ensure that users do not execute unauthorized commands?

- **Other Aspects**: ...
CoEdit: Our Approach

- Enhance the efficiency and performance through employing P2P technology.
  - Centralized and decentralized architecture
  - Direct communication between collaborative sites
  - Message routing in the collaborative group
  - Access control among different peers
CoEdit: Architecture

Collaborative Group 1

Collaborative Group 2

Internet/LAN

CoEdit Server

Collaborative Peer

Collaborative Peer

Collaborative Peer

Collaborative Peer

Collaborative Peer
CoEdit: Function Model

Collaborative User Interface Layer
- Dynamic Lock Mechanism
- Consistency Maintenance
- Co undo / Redo
- Co Design
- Co Awareness
- Co assistant
- Role Administrate

Collaborative Control Layer

Collaborative Transport Layer
Another Project

TrustedPeer --
One of the security related projects
TrustedPeer: Overview

Today’s Deployments Often Leave Clients Relatively Unprotected

Server
- Controlled physical access
- 24x7 monitoring/guard
- Highly regulated SW/HW configuration
- Intrusion detection SW
- Firewall
- Anti-virus
- Network segmentation
- Encrypted data
- UPS power protection
- Real-time monitoring
- Real-time backup
- Auditing & analysis tools
- Two-factor user auth.
- Configuration monitors

Network
- Encryption (IPSec, SSL)
- VPN
- Layered firewalls
- Intrusion detection SW
- 24x7 monitoring
- Network segmentation
- 802.1x (Radius)
- User authentication
- Two factor authentication
- Domain controllers
- Policy management
- Configuration monitors

Client
- Passwords
- Anti-virus
- User authentication
- Patch, Configuration, & Policy Control
- Intrusion detection SW
TrustedPeer: Overview

- Trust on client platform is needed in modern systems and emerging applications
  - Distributed Dissemination CONtrol (DDCON)
    - e.g., Health records of a patient may be transmitted from a primary physician to a consultant who can access them for some limited period of time and cannot transmit them to anyone else
  - P2P VOIP Application
    - Realtime protection of audio data in a platform
      - conversation is not eavesdropped or illegally recorded.
    - Forward control of audio object (e.g., voice mail)
      - Control the platform and user to forward
  - P2P E-Commerce
    - Electronic currency between peer platforms
    - Payment systems for p2p e-commerce
TrustedPeer: Overview

- Need new security model and architecture
  - Change of trust relation between client and server
    - No centralized and strongly protected server
    - Data located in peers or general client platforms
  - Location of policy enforcement changed
    - Client-side policy enforcement needs trust
  - Trust of platform and application
    - Dynamic environment
    - Software-based attacks
  - Trusted user authentication and authorization in client platform
  - Trusted path from peer to peer
    - Spoofing and “man-in-the-middle” eavesdropping or modification attacks
    - Trusted information exchange between peers
TrustedPeer: What's our Focus?

- Three types of researches for P2P security
  - Reputation systems
  - Recommendation systems
  - Trust systems

- TrustedPeer - part of the following project
  - Policy-based Secure Interoperability among Multiple Autonomous Domains
    - TrustedPeer: A Secure and Dynamic Trusted P2P System
    - OntoRBAC: Ontology-based Description and Enforcement of RBAC
    - OntoPolicy: Ontology-based Secure Interoperability among Multiple Security Policies (extended OntoRBAC)
Policy-based Secure Interoperability Architecture

Dynamic Trust Domain

Group A

Domain 1

Domain 4

Domain 2

Domain 3

Secure Negotiation

Secure Session

Distributed Network Environment

Group Security Policy

Local Security Policy

Global Security Policy
Protocol Layers

Control Layer
- Authentication
- Access Control
- Trust Management
- Policy Evaluation
- Policy Negotiation

Policy Layer
- Authentication Policy
- Authorization Policy
- Trust Policy

Mapping Layer
- Policy Representation & Transformation

Transport Layer
- Message Binding & Delivery

Auditing

Risk Assessment
OntoRBAC

- Kinds of methods for security policy description
  - Logic-based (FOL, Stratified Logic, Deontic Logic)
  - XML-based (XACML, XRBAC, SAML, ...)
  - Ontology-based (Rei, KAoS)

- OntoRBAC
  - Ontology-based Description and Enforcement of RBAC
  - Concepts: Entity, Subject, Role, PolicyRule, Permission, Action, Policy, ...
OntoRBAC: Concept-Relationship Diagram
**OntoRBAC: Architecture**

- **PEP**: Policy Enforcement Point
- **PDP**: Policy Decision Point
- **PAP**: Policy Administration Point

**Diagram**:
- **Access Requester**
- **Request + Credential**
- **Response**
- **Request**
- **Response**
- **Policy Loader**
- **Policy Engine**
- **OntoRBAC Policy**

**Network Flows**:
- Access Requester → PEP
- PEP → Access Requester
- PEP → PDP
- PDP → PEP
- PEP → PAP
- PAP → OntoRBAC Policy
- OntoRBAC Policy → PAP
- PAP → PDP
- PDP → PAP
- PAP → Policy Loader
- Policy Loader → Policy Engine
- Policy Engine → Policy Loader
- Policy Engine → PDP
- PDP → Policy Engine
- Policy Engine → PDP
- PDP → PAP
- PAP → Administerate
- Administerate → OntoRBAC Policy
- OntoRBAC Policy → Administerate

**Terminologies**:
- PEP: Policy Enforcement Point
- PDP: Policy Decision Point
- PAP: Policy Administration Point
OntoPolicy

**Goal**: Integration of Multiple Different Security Policies

- Heterogeneity of security model
- Heterogeneity of security policy (description)
- Heterogeneity of security semantics

**OntoPolicy**

- Ontology-based Secure Interoperability among Multiple Security Policies
- Nearly ongoing
Next ...

- Background of P2P Research
- A Glance of our Current Projects
- WebPeer: A Web Services Oriented Peer-to-peer System
- Experiments & Implementation
- Summary & Ongoing Work
Web Services are booming ...

- Classical Web Services
  - Service provider
  - Service requestor
  - Service broker (UDDI)
Disadvantages of Web Services

- Single Node Failure
- UDDI Bottleneck
- Limited Scalability
- Denial of Service (DoS) Attack
While Peer-to-peer Computing ...

- Sharing plentiful resources and services among network edges
- Federated cooperation among companies
- Having Lower costs of system maintenance
- Fault tolerance & load balance
Combination of the two technologies

- Using P2P-based technologies to publish and discover Web Services
  - Combination of centralized and decentralized characteristics
  - The node providing web services act as a peer
  - Each peer can request web services from other peers
  - Extend the reliability and scalability of the current web services architecture
Related Work

- Content-based search in P2P networks
  - CAN (AT&T), Pastry (Microsoft), Chord (MIT)

- Combination of web services and peer-to-peer networks
  - Self-Serv (UNSW, QUT)
  - Peermetrics

- P2P platform supporting Web Services
  - JXTA (Sun)
WSOP: Web Services Oriented Peer-to-peer Architecture
CSRB & LSRB

- **CSRB: Common Service Registry Broker**
  - Common Web Services
  - Service provider, requestor, broker
  - UDDI (CSRB), SOAP, WSDL

- **LSRB: Local Service Registry Broker**
  - Local Web Service
  - Peers (service provider, requestor)
  - Super peers (LSRB)
  - Peer group (same interests, neighbors)
Service Publishing

- Publishing services to LSRB
  - Register services to one peer group (super peer)
  - Register services to multiple peer groups

- Publishing services to CSRB
  - As a traditional service provider

- Publishing to both LSRB and CSRB
  - Mappings between LSRB and CSRB
Service Discovery

- **Step 1:** search the services in its peer group
  - If matches, enjoy the service
  - Otherwise, go to Step 2

- **Step 2:** request will be delivered to other peer groups
  - If matches, return the result, and cache the result on the way home
  - Otherwise, go to Step 3
Service Discovery

- Step 3: request will be delivered to CSRB
  - If matches, return the result
    - Establish mappings on LSRB
    - Cache the result on LSRB
  - Otherwise, service not found
Existing Problems

- Open problem:
  - Super peer will be the bottleneck if the peer group grows large enough

- Solutions:
  - Using more than one super peer in the group
  - Using Distributed Hash Table (DHT) to organize LSRBs on super peers
Using Multiple Super Peers

Peer Group 1

Peer Group 2

Peer Group 3
Modified DHT Approach

- Establish DHT among Super Peers
  - DHT: key-based search, need cost of maintenance
  - Super peers are less dynamic and transient than other peers
  - Publishing services
  - Discovering services

DHT: key-based search, need cost of maintenance.
Super peers are less dynamic and transient than other peers.

Insert(k1, v1)
Retrive(k1)
Another Solution

- Using peer groups, but no super peers
  - Publishing services to its neighbors
    - Selected peers (including itself)
  - Discovering services from its neighbors
    - Breadth-First Search (BFS)
    - Random Breadth First Search (RBFS)
    - Intelligent Search Mechanism (ISM)
    - Most results in the past (>RES)
    - Maybe other approaches ...
No Super Peers

IDC Lab, HUST
Service Discovery Techniques

Breadth First Search (BFS)  
- query all neighbors

Random Breadth First Search (RBFS)  
- query a random subset of neighbors
Intelligent Search Mechanism (ISM)

- intelligently query a subset of neighbors according to the relevance rank

Directed BFS and >RES

- query the neighbors that returned the most results in the last 10 queries
Next ...

- Background of P2P Research
- WSOP: Web Services Oriented Peer-to-peer Architecture
- WebPeer: A Web Services Oriented Peer-to-peer System
- Experiments & Implementation
- Summary & Future Work
Experiments

- Experiment Parameters
  - Vary nodes between 1,000..100,000
  - Vary percentage of nodes for service publishing between 0.1..0.5
  - Vary percentage of nodes for service discovery between 0.5..0.9
  - Initial topology random graph
Experiments

- Simulations for the four types of approaches
  - CSRB: Traditional UDDI server only
  - LSRB: Using single super peer
  - MSP: Multiple super peers
  - NSP: None super peers
  - BFS, RBFS, ISM, >RES

- Results will come out soon
Implementation: Detailed Architecture

Peer Node

- Web Services Configuration
- Security Policy
- P2P System Initialization
- Peer Group Discovery
- Peer Authentication
- Local Publishing
- Export Pipe
- Service Discovery
- Import Pipe
- Local Service Cache
- WSDL Parsing
- WSDL Generation
- WSIF Invocation
- WSDL Download
- Remote SOAP Server
- Local SOAP Server

Internet

P2P Overlay

Peer Group

Single Super Peer
An Instance of Web Services Advertisement

```xml
<?xml version="1.0"?>
<!DOCTYPE jxta:MSA>
<jxta:MSA xmlns:jxta="http://www.jxta.org">
  <MSID>urn:jxta:uuid-D110E7397F24401EA8318F383CFF294035B8C3C1CF6645EABD13B9C76EBB115906</MSID>
  <Name>JXTASPEC:WebServices:urn:helloservice</Name>
  <Crtr> Example.org </Crtr>
  <SURI> http://www.example.org </SURI>
  <Vers> 1.0 </Vers>
  <Desc> A service allow you to say hello </Desc>
  <Parm>
    <wsdlURI>
      http://www.example.org/helloservice/
      helloservice.wsdl
    </wsdlURI>
  </Parm>
  <jxta:PipeAdvertisement xmlns:jxta="http://jxta.org">
    <Id>urn:jxta:uuid-2EC8CDF870744C468B7CB111E337A01EE5E3818F9BBD405B90D2B7626E1549C504</Id>
    <Type> JxtaUnicast </Type>
    <Name> WebServices:RespPipe:urn:helloservice </Name>
  </jxta:PipeAdvertisement>
</jxta:MSA>
```
Algorithms of Discovering Web Services

PROCEDURE discoverWebServicesAdvertisement
BEGIN
    WHILE
        Looking up Web Services Advertisements in local service cache;
        IF (search result doesn’t match the request)
            BEGIN
                Sending discovery request to peer group;
                IF (search result matches the request)
                    Saving result in local service cache;
                ELSE
                    BEGIN
                        Sending discovery request to CSRB;
                        IF (search result matches the request)
                            Saving result in local service cache;
                        ELSE
                            Return result with no matches;
                        END;
                    END;
        END;
        return the discovery result;
    UNTIL (number of discovery request is 0);
END;

PROCEDURE getWsdlDocument
BEGIN
    WHILE BEGIN
        Getting a Web Services advertisement;
        IF (parameters of advertisement are not null)
            BEGIN
                Call doc(StructuredTextDocument) to create a structured text document;
                Saving parameters of advertisement into created document;
                elements := doc;
                WHILE BEGIN
                    Getting names of subitems;
                    IF (subitem is wsdlURI)
                        BEGIN
                            Getting the value of wsdlURI;
                            IF (the value of wsdlURI is not null)
                                BEGIN
                                    return the value of wsdlURI;
                                    break the inner LOOP;
                                END;
                            END;
                        END;
                END;
                END;
            UNTIL (peers has more elements);
    END;
END;

IDC Lab, HUST
WebPeer System Interfaces

IDC Lab, HUST
Next ...

- Background of P2P Research
- WSOP: Web Services Oriented Peer-to-peer Architecture
- WebPeer: A Web Services Oriented Peer-to-peer System
- Experiments & Implementation
- Summary & Ongoing Work
Summary & Ongoing Work

- **Security models** for different approaches and topologies
- **Semantic-based model** for service publishing and discovery
- **Possible mobility of Web Services** in P2P Environment
Thanks!