行政院國家科學委員會補助專題研究計畫

成果報告

發佈/訂閱式之 P2P 系統及其應用

計畫類別：☑ 個別型計畫  □ 整合型計畫
計畫編號：NSC 96－2221－E－009－142－
執行期間：2007 年 08 月 01 日至 2008 年 09 月 30 日

計畫主持人：袁賢銘
共同主持人：
計畫參與人員：葉秉哲，簡士強，林辰璞。

成果報告類型(依經費核定清單規定繳交)：☑ 精簡報告  □ 完整報告

本成果報告包括以下應繳交之附件：
☑ 赴國外出差或研習心得報告一份
☑ 赴大陸地區出差或研習心得報告一份
☑ 出席國際學術會議心得報告及發表之論文各一份
□ 國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、
列管計畫及下列情形者外，得立即公開查詢
☑ 涉及專利或其他智慧財產權，□ 一年□ 二年後可公開查詢

執行單位：國立交通大學資訊工程學系

中華民國 97 年 9 月 30 日
研究計畫中英文摘要：

（一）計畫中文摘要。（五百字以內）

網際網路的日益發達，程式之間透過網路來溝通及交換訊息也日趨頻繁。目前發佈／訂閱模式是很流行的中介軟體架構，主要的優點是：低耦合、非同步、合乎直覺的彈性運作模式，但當規模大到某個程度時，現有的實作卻有擴充、容錯方面的問題。另一方面，P2P 網路的特點在於：充分利用所有節點的資源、自行組織、負載平衡、容錯，但缺點是：如果使用者需要某個資料，必須透過查詢來表示，並且在每次的傳回結果中，都需要再進行過濾，才能得到此資料。

本研究計畫提出一套「發佈／訂閱式之 P2P 系統」，結合上述兩種網路系統的優點：藉由發佈／訂閱式系統提供上層的服務，而由 P2P 網路負責底層的訊息路由。如此一來，對於前者系統而言，底層的 P2P 網路提供了更有擴充性、自我組織容錯能力、更有效率的路由機制；對於後者而言，發佈／訂閱模式提供了更好的資訊過濾能力，能降低網路頻寬的損耗。

本計畫也從應用軟體角度提出一項Blog系統應用實例，以驗證系統的運作效益。

關鍵詞：P2P，發佈／訂閱式系統，中介軟體。
It is important for applications to exchange messages on the Internet. The publish/subscribe (pub/sub) model is very popular in current middleware architecture. The main benefits are: loose-coupling, asynchronous operation, intuitive and flexible operation. But current implementation suffers problems such as scalability and fault tolerance as the scale grows. On the other hand, the main benefits of peer-to-peer network (P2P) are: fully-utilized node resources, self-organization, load balance, and fault tolerance. But the main drawback is that query is needed whenever a user wants to find some data, and further filtering on contents obtained is also inevitable.

In this project we propose a “P2P pub/sub system” to incorporate the best of the two models: the upper service interface is adapted from the pub/sub system and the underlying message routing from the P2P system. Therefore, the synergy is that we will have better scalability and self-organized fault tolerance, and more efficient routing mechanism due to the P2P topology, and better message filtering capability and lower bandwidth consumption due to the pub/sub model.

In this project we also propose a real-world blog application to validate the effectiveness and efficiency of the platform.

*Keywords: P2P, pub/sub system, middleware.*
報告內容：

（一）前言

At present, the structured P2P algorithms have been proposed frequently. Consequently, the P2P application developers need to learn different API semantics. It generates additional efforts of switching to different P2P topologies. Moreover, it is difficult for the developers to evaluate the performance of an application based on a particular underneath P2P APIs. On the other hand, if the P2P framework can provide reusable and comprehensive network communication components, it can expedite developing progress. Therefore, the P2P protocols can easily accommodate to different network environments.

In this research, a novel P2P developing framework is proposed to assist in developing P2P applications by using various structured P2P protocols and P2P pub/sub algorithms. We design an architecture to construct the structured P2P functional blocks, including network communication components, P2P topology maintenance and routing, network bootstrapping, as well as pluggable pub/sub services.

（二）研究目的

In the application domain of content management system, e.g., personal blog system, large amount of information are created and requested over the entire user community. With the search capability, users can retrieve information which has particular contents according to given query. As the P2P community keeps advancing, however, the number of updating events will soon overwhelm the size of events that human can handle. By introducing pub/sub mechanism, applications can automatically disseminate information to the interested peers in P2P network. Like the RSS supported on many website, the pub/sub paradigm provides the functionality for users focusing on only the interested events. Therefore, pub/sub mechanism is an essential feature while designing a platform for developing P2P applications.

There are three aspects of developing a P2P Pub/Sub-related program: application developers, P2P protocol developers, and P2P pub/sub protocol developers. From the aspect of developing pub/sub applications, programmers usually need to learn new APIs when changing the underlying overlay network. The difference of semantics can reside in peer initialization, network construction, and even communication mechanism; that is, implementing the same functionality on different P2P APIs could cause code rewriting. The same situation happens in changing pub/sub APIs. This means application would be strong coupled with P2P and pub/sub implementations. Application developers have no chance to compare the performance of their systems on different overlays.
Issue 1a: Application developer need to learn different semantics from numerous P2P APIs.
Issue 1b: The cost of rewriting code is huge for testing performance of particular application on different P2P network.

For p2p pub/sub algorithm developers, the lack of a common platform for evaluating performance makes it hard to compare between algorithms. First, preparing identical test case on two different p2p pub/sub systems is cumbersome. Second, the delay of event dissemination needs to normalize due to the different implementation of internet communication.

Issue 2: P2P pub/sub algorithm developers need a common platform to compare with other algorithms.

While developing a P2P algorithm, developers writing their own code communicating with other peer through physical network connection. Each P2P API introduces redundant code on network programming. Developers take additional time on debugging network-related code. Without network-related code reusing, the effort for extending deployment environment is huge.

Issue 3a: P2P network developer write redundant code for network communication, make it hard to deploy P2P on different physical network environment.
Issue 3b: A common process is needed for overlay network initialization.

Our goal is to solve these issues mentioned above. Thus, a standardized API and communication mechanism for P2P application development is need to be defined.

（三）研究方法

Inspired by FreePastry and PeerSim, we further extend the P2P API by abstracting the physical network communication from P2P protocols and introduce additional bootstrapping facility. A standard pub/sub API is designed to accommodate with heterogeneous pub/sub model in pure P2P networks.
P2P Applications retrieve a live peer in the P2P network through Bootstrap Service. This live peer is used to initiate the join operation. Application can directly access the P2P Protocol Layer for message routing and performing lookup operation. By registering Pub/Sub Service to local peer, applications use Pub/Sub API to do event publication and subscription. P2P Protocol Layer delegates the physical network transmission to Transport Layer. Environment module loads external parameters from configuration file.

The pub/sub service module provides a light-weight API for executing pub/sub related task. Publisher and Subscriber define the common pub/sub API that can connect with arbitrary pub/sub service. Each Publisher and Subscriber is associated with one topic.

The PubSubService is a subclass of Service that defines the SPI needed for implementing P2P pub/sub algorithms. PubSubService receives the actions from pub/sub applications via Publisher and Subscriber. In order to accommodate to both topic-based model and content-based mode, the pub/sub API is designed with topic-based model and additional selector language like the one used in JMS for attribute filtering. The content-based model is also supported by introducing a wildcard topic. Pub/sub client program receives interested event via registering EventHandler.
The framework we proposed is based upon object-oriented architecture and event-driven methodology. According to the structured P2P specification, we enhance the functionalities into object models that fully describe the relationships between the identifier space and the routing protocol. Moreover, the framework proposed by Aberer et al., the additional service modules, e.g., P2P Storage Interface, and the P2P Basic Interface, i.e., P2P protocol, are objects directly inherited from the same parent class. However, in our architecture, we introduce pluggable modules, e.g., the Pub/Sub Service, those decouple from the P2P protocol implementation. The features are achieved by invoking services as events arrived. The events contain communication messages as well as topology modifications. The event of state transition of handling peer, i.e., peer joins to a network and peer is ready to receive message, is not propagated to the services. Developers can only perform stabilization and replication in proactive style while generating persistence services. Nevertheless, this pluggable approach makes a lightweight peer implementation. Therefore, the P2P Protocol Layer only needs to handle routing protocols. The additional pluggable services are independent modules not included in the layer.

Our design has been focusing on pure P2P networks. In other words, each peer in the architecture shares information and collaborates with other peers without a centralized server. In previous researches, publishers and subscribers are both clients of a message server. In our platform, each peer involves message dispersing and propagating via the pub/sub mechanism.
without an additional message server. Instead, each peer is involved in the information dispersal of the pub/sub mechanism in our framework, without establishing additional message server. The benefit of pure P2P is that applications do not depend on a pre-constructed server infrastructure. The index information is connoted in the P2P network topology and routing protocol, compared to the super-peer indexing mechanism used in JXTA. However, this statement assumes the computation power of each peer is about equal. According to the assumption, this framework does not grant developer the advantage deploying P2P applications on the environment of heterogeneous devices.

In previous research of P2P protocol, network bootstrapping is usually omitted. By considering the practicality of creating P2P applications, we define the bootstrap service interface and provide two boot server implementations. By externalizing the network communication, the framework allows different protocols transmitting messages through one single network port. With the evolving of CommunicationManager, the performance of all P2P protocols can be boosted.

The FreePastry library is an open source implementation of Pastry. With Scribe system implemented as an additional service, developers can create group communication system with efficient pub/sub capability. In the design of FreePastry, the factory methods are used for testing/simulating applications without modification to the source code. However, applications developed using FreePastry are limited to the functionalities that this P2P library offered, i.e., only Pastry network and Scribe system. Our framework provides a flexible architecture that application can easily deploy to any P2P network and any network environment. With the lightweight pub/sub APIs, application developers can adopt any P2P pub/sub service to meet their system requirements.

JXTA is a general P2P platform that allows heterogeneous applications be deployed on top of a virtual JXTA network. JXTA can provide additional structured P2P network functionality based on Peer Resolver Protocol. An open source project named Meteor [17] implements Chord and CAN on top of the JXTA platform. This approach deploys the DHT overlays upon the virtual JXTA network, which causes the performance downgrade because of the communication overheads among peers that introduced by JXTA. The JXTA platform provides a propagating pipe which can simulate pub/sub mechanism via the one-to-many message transmission. The message might be lost without noticed during the process of propagation. The performance degrading and reliability issue make this propagation mechanism not scaled to a large group communication system. In our framework, the message transmission among peers is directly delegated to physical network transportation, which does not incur the overheads of additional node discovery. Without message propagating, our pub/sub service can maintain a distributed multicast structure and support many-to-many message transmission. Therefore, disseminating information among peers will not cause unnecessary bandwidth dissipation.
（五）論文發表

Shih-Chiang Chien and Shyan-Ming Yuan, “Enhanced Search and Fault-Tolerance in Viceroy,” to be presented in 2008 海峽兩岸資訊科學與資訊技術學術交流會議。

Chen-Pu Lin, Yung-Wei Kao, and Shyan-Ming Yuan, “A P2P Blog System with OpenID Integration,” to be presented in 2008 International Conference on Convergence Information Technology.

Yung-Wei Kao, Hui-Zhen Gu, and Shyan-Ming Yuan, “Integration of face and hand gesture recognition,” to be presented in 2008 International Conference on Convergence Information Technology.

（六）畢業論文

簡士強，《點對點環境下的通用發佈／訂閱框架》，國立交通大學資訊科學與工程研究所碩士論文，2008/06。

林辰璞，《一個整合 OpenID 與點對點技術之部落格系統》，國立交通大學資訊科學與工程研究所碩士論文，2008/06。
計畫成果自評：

● 與原計畫相符程度

原計畫之構想規模及進度係以三年期來設計，現在由於計畫只通過一年，因此在探討規模方面，縮減為著眼於核心、基礎的 P2P pub/sub 機制上，在應用方面，也改成複雜度較低的 blog 系統。

● 達成預期目標情況

以核心 P2P pub/sub 機制而言，已大致達成原本設定的第一年目標。

● 研究成果之學術或應用價值

以本計畫設定的方向而言，是目前架構較完整、包容性較強的一套 P2P pub/sub 框架，具有初步的學術及應用價值。但還有許多細節課題有待進一步探討，才能更加完善。

● 是否適合在學術期刊發表或申請專利

需要再進一步探討未盡完善之處，並包容更多 P2P 協定、更多的應用實例，便具有期刊發表的價值。