International Journal of Grid Distribution Computing
Vol.8, No.2 (2015), pp.191-200
http://dx.doi.org/10.14257/ijgdc.2015.8.2.18

P2PCloud-W: A Novel P2PCloud Workflow Management Architecture Based on Petri Net

Xuemin Zhang, Zenggang Xiong*, Gangwei Wang, Conghuan Ye and Xu Fang

School of Computer and Information Science, Hubei Engineering University, Xiaogan city, Hubei Province, 432000, China
**jkxxzg2003@163.com*

Abstract

IaaS model in the Cloud Computing provides infrastructure services to users. However, the provider of such centralized Cloud requires notable investments to maintain the infrastructures. P2P Cloud, whose infrastructures are provided by multiple volunteer nodes in the P2P network, gives a low cost option to the provision of Cloud Computing. P2PCloud workflow is used to handle tasks with control and data dependencies, and plays an important role in P2PCloud environment. It is essential to design an appropriate model to satisfy the distributed and heterogeneous workflow systems that must be flexible, reusable, dynamic feature in higher level. In this paper, An architecture of a P2PCloud system firstly is proposed, and then a P2PCloud-W(P2PCloud workflow system) framework is proposed, which is a novel hybrid decentralized workflow management system facilitating both Cloud and P2P technologies. Finally, a P2PCloud-W case is analyzed using Petri Net.

Keywords: *P2PCloud, Workflow, Model, Architecture, Framework*

1. Introduction

Newly developing cloud computing [1] has brought about great benefits to both enterprises and individuals. With advanced technologies of virtualization and service, it incorporates various resources for user on-demand with open interfaces and transparent remote operations. While IBM, Google and Amazon are taking the lead in building general public cloud under the modes of SaaS (Software as a Services), IaaS (Infrastructure as a Service) and PaaS (Platform as a Service), many conglomerates have also obtained cost reduction and higher flexibility of resource sharing with the establishment of their own private cloud.

At present, Cloud Computing and peer-to-peer (P2P) are both hot topics respectively. However, the convergence of the two systems is increasingly visible: the two research communities started to acknowledge each other by forming multiple research groups that study the potential lessons that exchanged. P2P research focuses more and more on providing infrastructure and diversifying the set of applications; Cloud research is starting to pay particular attention to increasing scalability.

Cloud Computing and P2P systems enable users to share resources across the traditional domain boundaries. P2P offer another form of loosely coupled resource sharing at a very large scale. P2P systems exhibit a different set of characteristics like decentralization and distributed control. Recent research efforts explore the idea of combining approaches from both P2P and Cloud systems to harness the benefits of both worlds. Up to now, to achieve good load balancing, robust and scalability, many cloud computing models have been proposed, which are based on P2P technology. Such systems are called P2P Cloud [2].

* Corresponding Author

ISSN: 2005-4262 IJGDC

Compared with traditional workflow systems, Cloud workflow plays an important role in Cloud computing, due to the following advantages:

(1) The strong capability to dynamically congregate the distributed and heterogeneous resources to compose the new application;

(2) The management mechanism can facilitate the internal cooperation with each other among the different organization;

(3) It can make use of the resources distributed specific domain to improve throughput or reduce the implementation price of the system;

(4) It can bestraddle some management domains to obtain the particular capability of attending to or settling matters;

(5) It involves the integration for many groups of different parts in management workflow system.

However, conventional Cloud workflow systems, which adopt the traditional client-server architecture for centralized coordination and control of the business management processes, will exhibit weaknesses such as inflexibility, limited scalability, poor robustness, and insufficient openness in dealing with semantic Cloud enabled collaborative design problem due to its inherently decentralized nature. A P2P workflow, on the other hand, facilitates P2P technologies to workflow for direct communication and cooperation among relevant peers. The P2P workflow systems abandon the centralized data repository and control engine and fulfill the whole workflow functions by distributing both data and control. Thus the performance bottlenecks are likely eliminated and the system scalability can be greatly enhanced.

Therefore, to tackle the shortcomings of conventional Cloud workflow systems, this paper describes a preliminary attempt at decentralized Cloud workflow management with a P2P-based multi-agent framework for collaborative design, as agent technology is considered as an intelligent solution that provides autonomous and flexible problem solving capabilities while P2P technology is considered as a decentralized, scalable solution that deploys the multi-agent system in dynamic and open P2PCloud environments.

In this paper, we firstly give a P2PCloud resource management system architecture, and then we propose a P2PCloud workflow system framework, then a P2PCloud-W(P2PCloud workflow system) framework is proposed, which is a novel hybrid decentralized workflow management system facilitating both Cloud Computing and P2P technologies. Finally, a P2PCloud-W case is analyzed using Petri Net.

The rest of this paper is structured as follows. Section 2 surveys the related work. Section 3 describes the P2PCloud resource management system architecture. Section 4 proposes P2PCloud workflow system framework. Section 5 illustrates the application of P2PCloud Workflow System Case and finally, Section 6 concludes the paper and plans for future work.

2. Related Work

Cloud workflow is an important cloud service, which contributes to enhance the efficiency of Cloud task execution and reduce the cost. Cao, *et al.,* [3] adopted two-level handling framework when implementing the GFMS (Grid workFlow Management System). Hwang, *et al.,* [4] introduced fault tolerance in the framework of GFMS. Yu, *et al.,* [5] adopted two hierarchical schemes of centralized task management and distributed task processing in the engine of GFMS. The above frameworks of GFMS were built on the general WFMS (Work-Flow Management System) and mainly focused on the workflow implementation, but they considered less the bottom generic modeling and were difficult.

Georgios John Fakas [6] presents the architecture of a novel Peer to Peer (P2P) workflow management system. The proposed P2P architecture is based on concepts such

as a Web Workflow Peers Directory (WWPD) and Web Workflow Peer (WWP). The paper is also mainly focused on the workflow implementation. Jun Yan [7] proposes an innovative, decentralized workflow system called Swinburne Decentralized Workflow (SwinDeW), which is a typical p2p based workflow system.

The above workflow systems only adopt the workflow frameworks or the P2P workflow frameworks. This paper described a workflow framework for P2PCloud environment and gives a performance evaluation.

3. P2PCloud System Architecture

Generally, the P2PCloud system architecture is composed of many supernodes [8]. Each CloudPeer represents a super management domain. Each CloudPeer controls the access of a group of local computing resources. It plays two roles: one is as the resource provider, allowing its (or local others) free resources to implement the other supernodes' resources; the other is as a consumer, arbitrarily uses the local resources or the free resources of other supernodes to carry out its task. The idea of P2PCloud system architecture proposed in this paper comes from [8], but it differs from its idea. As shown in Figure 1, the bottom communities of the model using the traditional cloud technologies, and the P2P mode is adapted to interact information between CloudPeers.

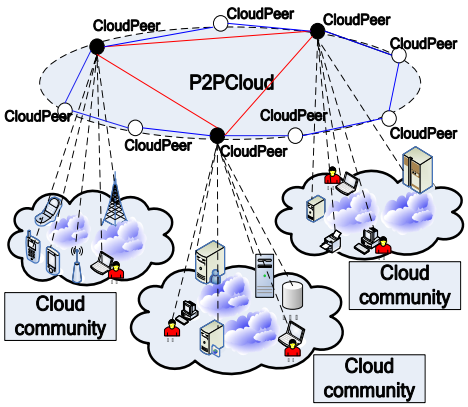

Figure 1. P2PCloud System Architecture

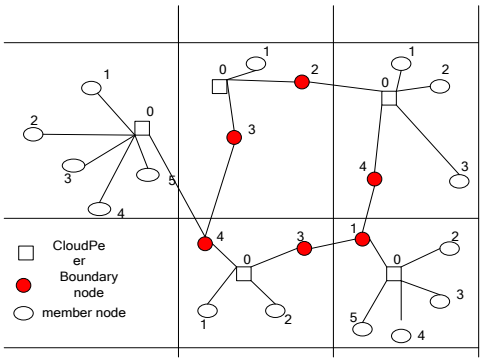

Figure 2. Node Connected Model in P2PCloud

Here, CloudPeer is equivalent to a super node. When they search resources, the users firstly query the resources in the domain of CloudPeer. If no query result, the search will be carried out through CloudPeer to query the other CloudPeer with P2P mode. We give some definitions as follows:

In this paper, the interconnected Cloud service nodes with same geographical location are divided into Cloud community, and then one representative node in each Cloud

community is selected, which is called CloudPeer The joint node between Cloud communities is called boundary nodes. In a Cloud community, the nodes except the CloudPeers and boundary nodes are called member nodes, as is shown in Figure 2.

Definition 1 (Cloud Communities): In Cloud layered structure, the interconnected service nodes in cloud are divided into a number of reciprocal groups (RG), with each RG as a Cloud community.

Example 1, As is shown in Figure 2, the Cloud environment is composed of five Cloud communities-A, B, C, D,and E. Cloud Community A is composed of 5 member nodes and 1 information node. Cloud Community B is composed of 1 member node, 2 boundary nodes and 1 information node.

Definition 2 (CloudPeer): For a Cloud community, there is a CloudPeer, which is in charge of the resource management of this Cloud community, including the updating, registering and deleting of the resources. Maintaining a resource list in an CloudPeer is Similar to the literature, and, when Cloud community members need to search resources, they first find out whether that resource was registered in the resource list, and if registered, they will locate the target nodes according to the registration; if not, the request will be transferred by the CloudPeer and the GAA algorithm in [9] will be run to find out the shortest path reaching the required resource node.

Example 2, In Figure 2, the node with the serial number 0 is as the CloudPeer in each Cloud community. Suppose A.0 is the representative of the nodes in Cloud community A.

Definition 3 (Boundary Nodes) The joint nodes between the Cloud networks are known as the boundary nodes, and the Cloud community X has at least one external link cross the community border.

Definition 4 (Simple Connection Undirected graph): Suppose G=(V,E), G is called simple connection

undirected graph, and only when Graph G satisfies the following two conditions: 1) G is free of self-ring, and is a connected undirected graph; 2) There is at most one border between any two nodes.

Definition 5 (Neighboring Nodes): If r and s are two arbitrary nodes in graph G=(V,E), that is, r, s∈V , and if there is a border in E linking the two nodes r and s, then nodes r , s are called neighboring nodes to each other.

4. P2PCloud Workflow System Framework

This paper proposes P2PCloud workflow system framework (named P2PCloud-W) as shown in Figure 3.

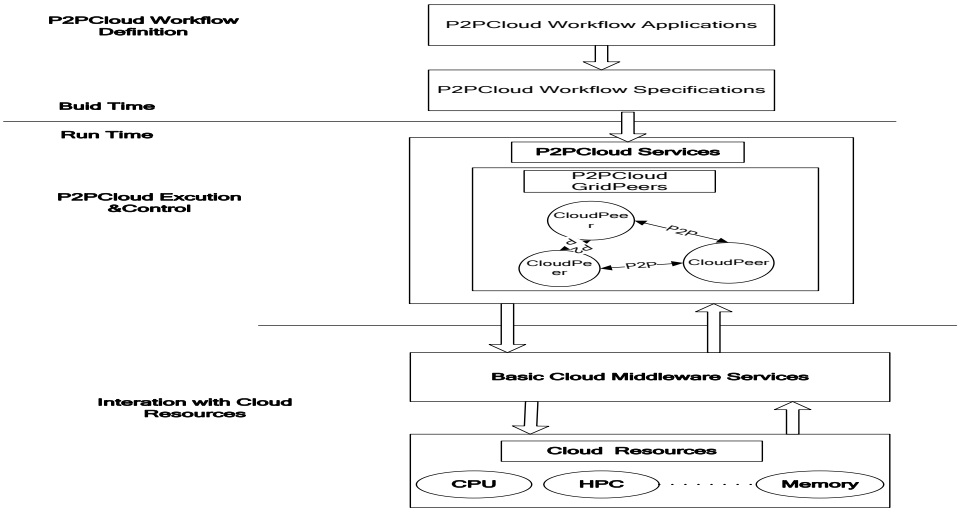

Figure 3. P2PCloud Workflow System Framework

As we mentioned before, P2PCloud-W is a P2P-based Cloud workflow system that enables workflows to be executed over a Cloud environment using direct P2P communications among Cloudpeers. This is achieved by wrapping P2PCloud-W Cloudpeers inside Cloud services and deploying them as Cloud middleware applications. This relationship can be seen in Figure 3. Once deployed, P2PCloud-W Cloudpeers will search for and connect with other P2PCloud-W Cloudpeers. After that, the Cloudpeers use P2P to exchange various information required to execute a workflow.

Unlike traditional Cloud workflow system, a Cloud node runs as a Cloud service along with other Cloud services. However, P2PCloud-W communicates with other Cloudpeers, a platform for P2P communication. As Figure 4 shows, a P2PCloud-W Cloudpeer consists of the following components:

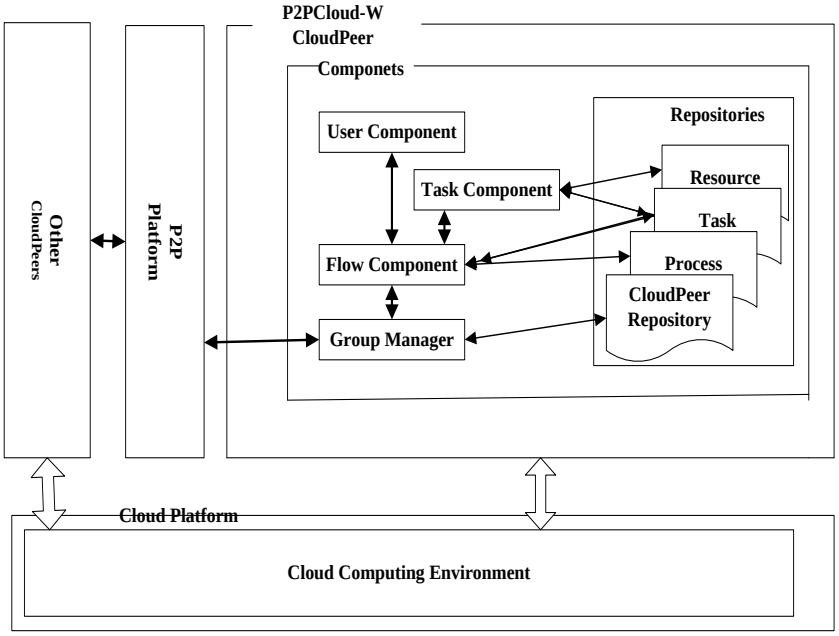

Figure 4. Architecture of P2PCloud-W Cloudpeer

The Task Component manages the P2PCloud workflow tasks. It has three main functions. First, it provides necessary information to the Flow Component for scheduling and stores received tasks to Task Repository. Second, it determines the appropriate time to start, execute and terminate a particular task according to the capability. A capability in P2PCloud-W is an object encapsulating rules with a role in workflow processes, which include the responsibility of this role, usage scenarios of this role, application-related constraints of each scenario (input, allowable operations, output, *etc.*), and so on. The resources that a workflow task instance may require are stored in the Resource Repository.

The Flow Component interacts with all other modules. First, it receives the workflows definition and then creates the instance definition. Second, it receives tasks from other peers or redistributes them. Third, it decides whether to pass a task to the Task Component to execute locally or distribute it to other peers. The decision is made according to the capabilities and load of itself and other neighbors. And finally, it makes sure that all executions conform to the data dependency and control dependency of the process definitions which are stored in the Process Repository and the Task Repository.

The Group Manager is the interface between the CloudPeer and P2P platform. In P2P platform, all communications are conducted in terms of CloudPeer group, and the Group Manager maintains the CloudPeer groups the CloudPeer has joined. The information of the CloudPeer groups and the CloudPeer in them is stored in the CloudPeer Repository.

While a P2PCloud-W CloudPeer is implemented as a P2PCloud service, all direct communications between CloudPeers are conducted via P2P. CloudPeers communicate to distribute information of their current state and messages for process control such as heartbeat, process distribution, process enactment etc.

The User component is the interface between the corresponding workflow users and the workflow environment. In P2PCloud-W, its primary function is to allow users to interfere with the workflow instances when exceptions occur.

5. P2PCloud Workflow System Case Study Analysis

In this section, we facilitate a case study to illustrate how P2PCloud-W supports the execution of P2P based Cloud workflows.

P2PCloud -W Procurement Scenario. To illustrate the problem, consider the example presented by Figure. 5, involving four business partners (a customer, a producer and two suppliers) illustrating the three cooperation phases mentioned above. The Customer sends an order for a product. Then it receives a notification announcing that the product has been taken into account by the producer. When the product is ready, the customer receives the delivery and then the invoice. Finally, he pays for the product he has ordered. The producer, waits for an order request. Then he searches for two suppliers to provide him with needed components in order to satisfy the received order. After that, he notifies the customer that his order is taken into account and waits for the suppliers response. When he receives the requested components, he assembles them and delivers the product to the customer. Finally, he sends the invoice and waits for the payment. The last partners are the two suppliers which, in our example and for simplicity purposes, have exactly the same workflow. When a supplier receives an order, he begins by producing it and then checking it. If the product conforms to the specification, it will be sent to the requester, otherwise another product must be produced and the process is repeated until the order is satisfied.

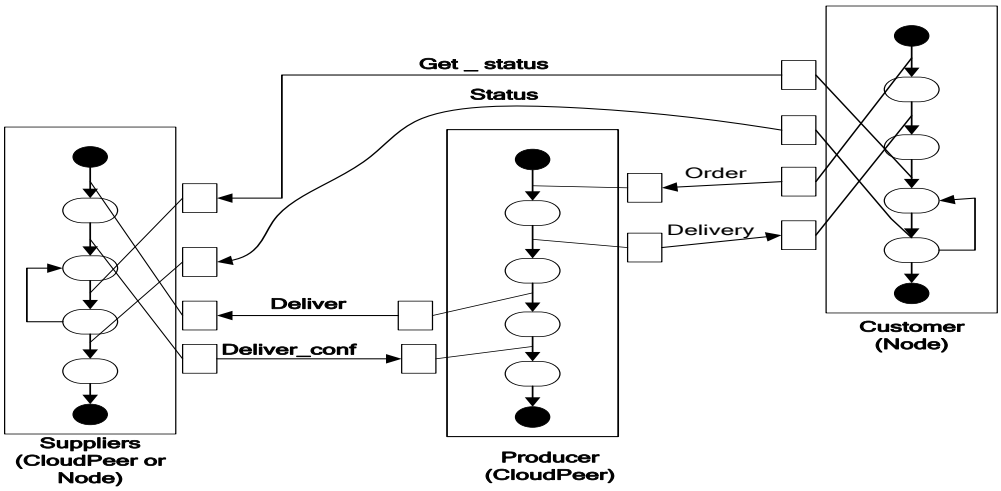

Figure 5. P2PCloud-W Procurement Scenario

Figure 5 represents the P2PCloud -W global relationships, but not the local workflows of the P2PCloud Workflow parties involved. In the following, the P2PCloud workflow net model is used due to the conformance to addressing properties of inter-CloudPeer workflows constructed from local workflows.

Local P2PCloud Workflow Model. A Petri net which models the process aspect of a workflow, is called a P2PCloud WorkFlow. It should be noted that a P2PCloud workflow net specifies the dynamic behavior of a single case in isolation.

Definition 6 (P2PCloud Workflow): A Petri net PN=(P,T,F) is a P2PCloud Workflow if and only if :

PN has two special places: I and o. Place I is a source place: $\bullet i = \phi$. Place o is a sink place: $o \bullet = \phi$.

If we add a transition $t^* \bullet = \{i\}$, then the resulting Petri net is strongly connected.

A P2PCloud Workflow consists of places (circles) representing business tasks and transitions (squares) connecting places representing a message exchange. The transitions are labeled with s#r#msg representing sender s and receiver r of the message as well as its message name msg. P2PCloud Workflow Nets contain a single final place represented by a circle with a solid line within the graph. A token is depicted as a dot within a circle. A transition is enabled if all input places of a transition contain a token. If a transition is enabled, it might fire, that is removing tokens from incoming places and inserting new tokens to outgoing places of the transition. The current distribution of tokens over the places describes the actual status of the workflow and is named marking.

The local P2PCloud workflows of the parties involved are described in Figure 6 forming the global P2PCloud workflow described above and guaranteeing a successful execution of it. The process is started by the customer C sending a C#P#order message to the Producer P, which informs the Supplier S via P P#S#deliver message to deliver the ordered goods. The Supplier S confirms this request (S#P#deliver_conf message) to the Producer P, which forwards the delivery details of the order (P#C#delivery message) to the customer C. Afterwards, the customer C is allowed to do parcel tracking directly with the Supplier S by sending P C#S#get_status message answered by P S#C#status message. Finally, the customer and Supplier process is terminated by the customer C sending P C#S#terminate message.

Global P2PCloud Workflow Model. The global P2PCloud workflow can be constructed by relating two transitions labeled equally and owned by different parties via an asynchronous channel. Each channel finally results in an additional place connected by an incoming arc with the “sending” transition of a message and an outgoing arc to the” receiving” transition. A token located in a newly introduced place can be interpreted as a message contained in the channel waiting for being received by the corresponding party. Finally, to fulfill the P2PCloud workflow constraints of a single token as an initial marking and a single finite place, new initial and final places are introduced connected to the previous ones by new transitions t_{start} and t_{final} distributing tokens to the local start places and collecting tokens from local final places respectively.

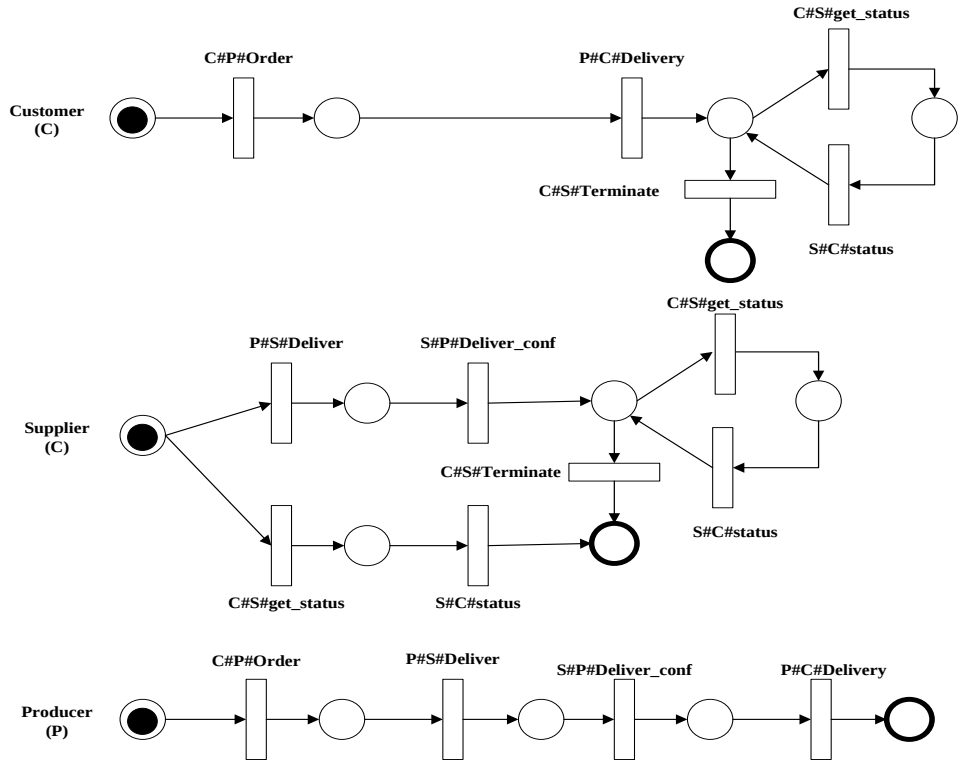

Figure 6. Local P2PCloud Workflow Model

The global P2PCloud workflow of above example is described in Figure 7, where the original local P2PCloud workflows are emphasized by the grey boxes with the customer local P2PCloud workflow being on top followed by the producer local P2PCloud workflow, and finally the supplier local P2PCloud workflow.

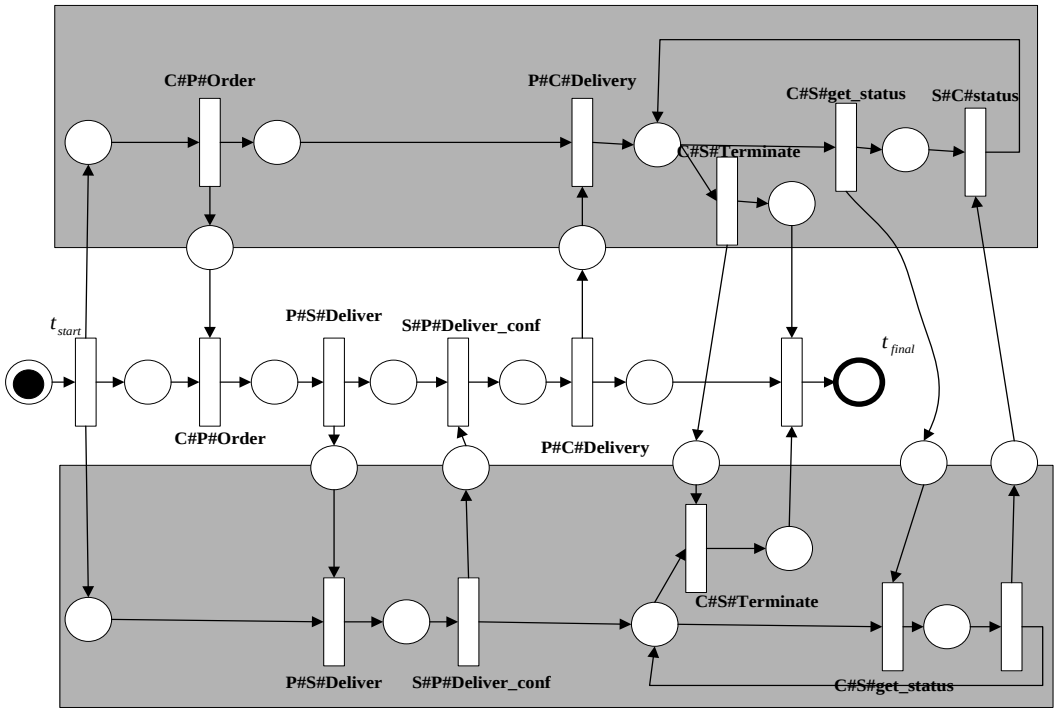

Figure 7. Global P2PCloud Workflow Model

6. Conclusions and Future Work

In this paper, we firstly present P2PCloud system architecture, then gives the P2PCloud workflow system architecture. Finally, we describe the P2PCloud workflow system model (P2PCloud-W) using Petri Net. In the future, a P2PCloud workflow prototype will be constructed and P2PCloud -W still needs further improvement. On one hand, load balancing would occur when multiple Cloudpeers have the same capability. However, the task scheduling is now primarily performed at the task instantiation stage and static in the current system which is insufficient. New scheduling algorithms will be developed to balance the load which is of course in a dynamic and distributed manner. On the other hand, monitoring is not implemented in the P2PCloud workflow system. However, it would be desirable to be able to monitor the status of the workflows. In addition, P2PCloud will also be compared with other Cloud workflow systems.

Acknowledgements

This research is partially supported by the National Science Foundation of Hubei Province (No.2014CFB188, No. 2013CFC005), Hubei Provincial Department of Education Outstanding Youth Scientific Innovation Team Support Foundation (T201410).

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "Above the Clouds: a Berkeley View of Cloud Computing", University of California, Berkeley, **(2009)**.
[2] O. Babaoglu, M. Marzolla and M. Tamburini, "Design and Implementation of a P2P Cloud System", Proceedings of 27th ACM Symposium on Applied Computing, **(2012**) October 15-19, Trento, Italy.
[3] J. Cao, S. A. Jarvis, S. Saini and G. R. Nudd, "Gridflow: Workflow management for grid computing", Proceedings of the 3rd IEEE/ACM International Symposium onCluster Computing and the Grid, **(2003)** May 11-15, Tokyo, Japan.
[4] S. Hwang and C. Kesselman, "Grid workflow: A flexible failure handling framework for the grid", Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing, **(2003)** June 22, Seattle, USA.
[5] J. Yu and R. Buyya, "A novel architecture for realizing grid workflow using tuple spaces", Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing, **(2004)** November 8, Pittsburgh, USA.
[6] G. J. Fakasa and B. Karakostas, "A peer to peer (P2P) architecture for dynamic workflow management", Information and Software Technology, vol. 46, **(2004)**.
[7] J. Yan, Y. Yang and G. K. Raikundalia, "SwinDeW– A Peer-to-peer based Decentralized Workflow Management System", IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 5, no. 36, **(2006)**.
[8] X. Zeng-Gang and Y. Yang, "Research on two-phase grid task scheduling based on Petri nets Journal on Communications", vol. 8, no. 30, **(2009)**.
[9] Z. Xiong, Y. Yang, X. Zhang, *et al.*, "Integrating Genetic and Ant Algorithm into P2P Grid Resource Discovery", Proceedings of the 3th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, **(2007)** October 11-13, Kaihsiung, Taiwan.

Author

Xuemin Zhang, received the the Bachelor degree in computer science from Hubei Normal University, China, in 2001, and the MA degree in computer science from Wuhan University of Technology, China, in 2009. She is now an associate professor in Hubei Engineering University. Her research interests are in the areas of Cloud computing, distributed systems, Service Computing. She is a member of the IEEE and the ACM.