Web Service-Driven Framework for Maintaining Global Version Consistency in Distributed Enterprise Portal

Hui-Ling Lin[1], Shao-Shin Hung[2], Derchian Tsaih[3] and Chiehyao Chang[4]
[1]*Department of International Business, Ling Tung University, Taiwan, R.O.C*
[2]*Department of Computer Science and Information Engineering, WuFeng Institute of Technology, Chiayi, Taiwan, R.O.C*
[3,4]*Department of Electronic Commerce Management, Nanhua University, Chiayi, Taiwan, R.O.C*
[1]*ljoyce@mail.ltu.edu.tw*; [2]*hss@cs.ccu.edu.tw*; [3]*dtsaih @mail.nhu.edu.tw;*
[4]*cychangs@mail.nhu.edu.tw*

Abstract:- The explosion of the web has led to a situation where a majority of the traffic on the Internet is web related. Today, practically all of the popular web sites arc served from single locations. This necessitates frequent long distance network transfers of data (potentially repeatedly) which results in a high response time for users, and is wasteful of the available network bandwidth. This paper presents a new approach to web replication, where each of the replicas resides in a different part of the network, and the browser is automatically and transparently directed to the "best" server. This paper presents a transnational hierarchical global patch consistent model called THGPCM (Transnational Hierarchical Global Patch Consistent Model). Improving the OPDS (Original Patching Data Source) enabled network equipments, capable of updating patch parameter that existed in enterprise with specified OPDS partially dependency relationships. Apply scenario can reduce the global patch service cost of transnational enterprise network equipments and minimum the turnaround time of patch service delay.

Keywords: Service-oriented, patch, consistency, routing, distributed

1 Introduction

The explosion of the web has led to a situation where a majority of the traffic on the Internet is web related. In fact, in the beginning of 1995, web traffic became the single largest load on the Internet [1,16]. Today, practically all of the popular web sites are served from single locations. This necessitates frequent long distance network transfers of data (potentially repeatedly) which results in a high response time for users, and is wasteful of the available network bandwidth. Moreover, it commonly creates a single point of failure between the web site and its Internet service provider.

On other side, The rapid evolution of information technology has created opportunities to offer various software tools, protocols and standards to support e-learning where learners can engage and manage their learning. As a result, learners demand effective personalized learning support that facilitates them to achieve their learning goals, rather than merely attend learning materials that are largely pre-prepared for them. This significant change in learning requirements imposes a new learning paradigm which enables flexible mode of content configuration, adaptive delivery and effective assessment [9,14].

We present a new approach that uses web replication where each of the replicas resides in a different part of the network. The key contribution of this paper is how to have the client - the web browser - automatically and transparently contact the *best* replica, taking into account:

I. Network topology: which replica is "closest" to the client, network-wise.
II. Server availability: which servers are currently active.
III. Server load: which server is currently able to return the most rapid response.

Most of the existing web replication architectures involve a cluster of servers that reside at the same site. These architectures improve performance by sharing the load between the different replicas, and improve availability by having more than one server. However, they cannot address the performance and availability problems embedded in the network

Service-oriented architectures (SOA) are an architectural style in which all functions, or services, are defined using an interface description and have invokable, platform-independent interfaces that are called to perform business processes. Unfortunately, the design principles and proven practices (also referred to using the term “best practices”) behind process-driven SOAs have not been well documented so far [1, 12]. On the other side, new services and applications demands increase growth in future generation network and gradually leading to the concept of extensible enterprise portal (EP). Deploying new services within an EP in a dynamic and flexible manner requires open architectures for controlling new data flow and web service processing functionality at run time. Traditional EP, has monolithic architectures which limit flexibility and extensibility for supporting dependency copies consistency issues. In other words, they always have the following problem — in large scale distributed systems, caching and replication could greatly speedup access and increase availability of distributed shared objects’. Both caching and replication create multiple copies of the state of an object which introduces the problem of *consistency* among the copies. The performance and functionality of such a distributed system could depend very much on the level of consistency it provides. Consistency of replicated state can be guaranteed by forcing operations to occur in the same order at all EPs.

Several studies are currently focusing on the design and concept of using caching, middleware and database replication technologies [1-3, 13-15] to provide modular and flexible EP to meet the requirements. However, existing EP architectures do not meet the efficiency and scalability global data consistency.

This paper presents the design, implementation of an extensible, scalable and distributed hierarchical based routing portal, called *THGPCM* (Transnational Hierarchical Global Patch Consistent Model), capable of supporting *ripple-and-propagate* updating copies that exist in internal heterogeneity of database with specified OPDS (Original Patch Data Source) partially dependency relationships. The paper also proposed a new scalable replica routing policies that dynamic reconfiguration distributed forwarding data path in hierarchical enterprise databases.

However, wide area global dependency patch replica consistency in TDHEO （Transnational Distributed Hierarchical Enterprise Organizations） become increasingly important and popular. To meet this aim, our approach can be divided into 2 phases. First, a novel global dependent replica updating platform was proposed in the first phase, which consists of active, real time, automation, and global FPC (Forwarding Portal Clusters) web services routing technologies. When OPDS object was triggered by update event, all patch replica with the specific OPDS dependent relationship could be automatically and real time updated. Second, the following phase proposes a specific OPDS mechanism with dependent patch replica consistency maintenance scheme; i.e., a framework that provides update routing policy with adaptive and adjustable properties — like update reserve priority, and value added update services, based on usage mining results of patch replica with specific OPDS dependency relationships.

Based on the core of the ubiquitous updating process framework, several models were built. The usage mining rate for OPDS dependency patch replica is a function of dependency replica operation response time and turnaround time. The updating route policy scheduling rate is a function of usage mining and reservation priority. The production rate of activity represented as a time variable. Different value added update service schema was also analyzed. For example, active updating pushed by real time, periodic, reservation, and queuing with priority / degradation / preemption etc. Apply scenario can reduce the global patch service cost of transnational enterprise network equipments and minimum the turnaround time of patch service delay.

Section 2 of this paper summarizes related work. In Sect. 3-1, the active OPDS dependency copies wide area consistency processing models are presented. In Sect. 3-2, we illustrate a OPDS agents [4-7] and web service [8-15] driven architecture. In Sect. 3-3, we illustrate the agent through a variety of flexibility and customization message publishes mechanism to transfer OPDS update copy. In Sect. 3-4, Forwarding EP (Enterprise Portal) agent use pipeline mechanism automatically conversion of all kinds of dependency patch replica format and update TDHEO patch replica with OPDS dependencies. In Sect. 3-5, we show the system sequential diagram. The OPDS hierarchical ripple propagation routing algorithm discussion is given in Sect. 4. Finally, the distribution of web service processing functions over distinct and heterogeneous entities must be achieved according to capabilities. Current cluster based EP do not provide these needed mechanisms and architectures to create a distributed data path composed of multiple web service processing functions located in different clusters. This is exactly the aim of THGPCM which provides scalable and high performance replica forwarding path using the optimization framework proposed in Sect. 5.

2 Related work

Nowadays, generic Internet technologies such as XML are being adopted to manage network resources. Thus in this background, XML-based network management, which applies XML technologies to network management, has been regarded as an alternative to existing network management. Extensible Markup Language (XML) is a meta-markup language standardized by the World Wide Web Consortium (W3C) for document exchange in the web. XML is now a standard that is supported and accepted by thousands of vendors as well as a lot of related technologies and tools. Under this background, Web services have also been emerging as a promising Internet-oriented technology and architecture for network management [17-27]. A number of widely adopted Web services technologies are now available, such as Simple Object Access Protocol (SOAP) [15] [16] [17], Web Services Description [24] [25] [26] [27], and Universal Description Discovery, and Integration (UDDI) [22]. The word "services" in Web services refers to a Service-Oriented Architecture (SOA). In an SOA, fiinctionality is "published" on a network where two important capabilities are also provided - "discovery", the ability to find the functionality, and "binding", the ability to connect the functionality. So when considering a SOA, these three parts must be take into account, which are briefly presented as "publish", "find", and "bind".

The concept of a SOA is not new, for service-oriented architectures have been used for years. However, what is relatively new is the emergence of Web services based SOAs. In a Web services-based SOA, three important roles are Web service provider, Web service requester, and Web service register, which correspond to the "publish", "find", and "bind" aspects of a SOA.

Today's global patch consistent solution have been proposed of following kinds: data Integration middleware [12][13], xml-based integration middleware, xml-based data integration platform and xml-powered integration middleware. All these data integration system that enables enterprises to rapidly build web services and applications that can query multiple, disparate data sources and provides a unified result. That also provides direct access to business information which was previously cumbersome, time-consuming, or cost-prohibitive to acquire

Enterprises must pay more effort to keep global network equipment patch consistent than before, but the original version of patch replica dataset is locked up in a patchwork of disparate systems : legacy systems, relational databases, flat files, document repositories, data warehouses, Microsoft excel files, web pages, email and more other data sources. Beyond a company's firewall is an ocean of additional data in partner, customer and supplier systems. Because these systems were never designed to work together, it's extremely challenging to take full advantage of the information locked up in all that data. Yet competitive market requires that companies do the job exactly.

If patch replicas having partial dependency relationships with specific OPDS object inside TDHEO can't keep mutual consistency, the following problems will be happened. 1. The Bullwhip Effect will be occurred in dependency patch replicas flow chain operation. 2. Enterprise network equipments can't make sure the newest version patches. 3. It is easy for TDHEO knowledge base to occur horizontal information lacks harmony. 4. The OPDS can't control its patch replicas updating flow, transaction, usage mining results and it can't support different priority and value added updating services. 5. The dependent patch replicas usage mining results can't be sent to OPDS site to make adaptive patch replicas updating route policies and value added updating services.

3 Concept model description

3.1 THGPCM model of patch consistency

In this paper, a novel framework was proposed which consists of active, real time, automation, and global routing patch replica from OPDS domain to TDHEO set. See in Figure 1.

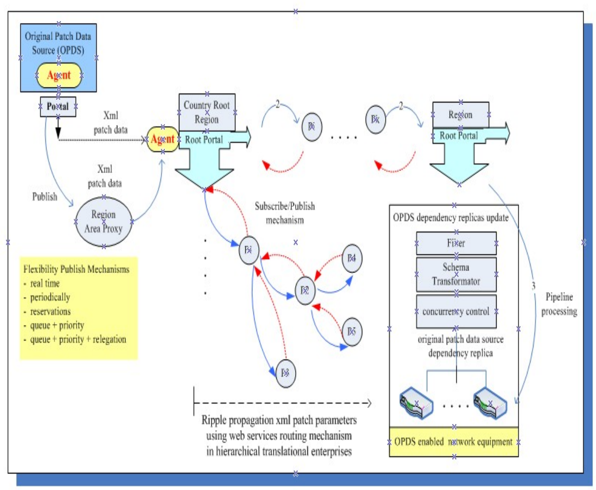

Figure 1. THGPCM concept model

When OPDS object occurred add/insert events, according to subscribe lists. New copies will propagation push to all global TDHEO portal which contained dependency patch replica. The EP then make one of the choices about discard/rerouting to lower hierarchy portal, automatic pipeline processing and schema transformation. All wide area OPDS dependent patch replica consistency and correctness can be automatic maintenance. Enterprise intranet application system need not change any source code. Infrastructure Model was shown in Figure 2.

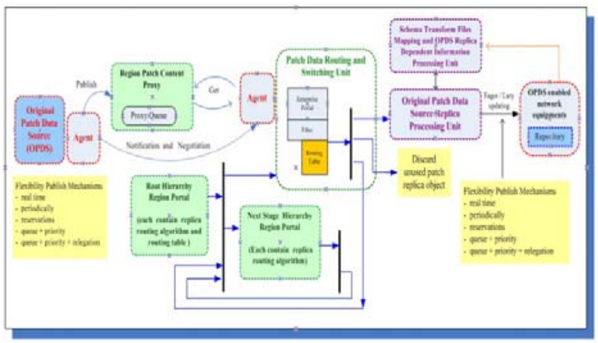

Figure 2. THGPCM infrastructure model

3.2 Enterprise portal model

Application operation patch replica in intranet have three kinds of models. In Figure 3, patch replica with partial specific remote OPDS dependency relationships were denoted by (a). Patch replica with OPDS role in extranet enterprise flow chain was denoted by (b). A replica with partial dependency with a specific self-enterprise intranet OPDS was denoted by (c). Each EP can be two roles : FP (Forwarding Portal) and FPC (Forwarding Portal Cluster).

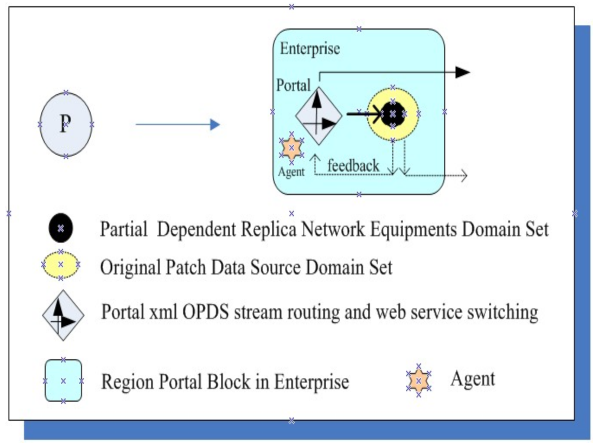

Figure 3. enterprise portal model

3.3 OPDS ripple propagate routing mechanism

In this section, which consists of OPDS subscribe and ripple propagation message chain shown in Figure 4. and 5. When OPDS object was triggered by update event, the message chain was propagation in active, real time, and automation. Then all patch replica with the specific OPDS dependent relationship could be globally updated.

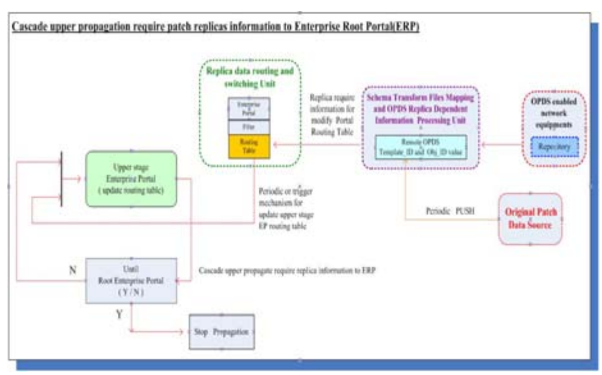

Figure 4. OPDS ripple propagation request info.

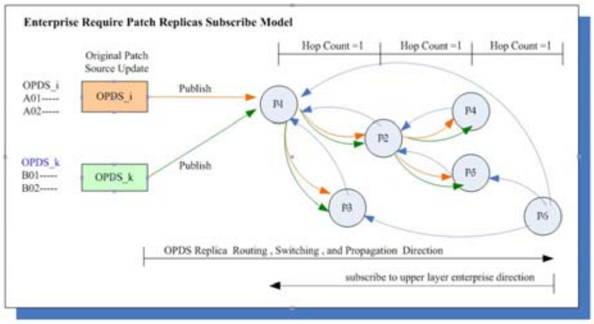

Figure 5. OPDS ripple propagation update chain

3.4 Portal agent pipeline conversion schema

The EP then make one of the choices about discard/ reroute to lower hierarchy portal/automatic pipeline processing and schema transformation. See in Figure 6

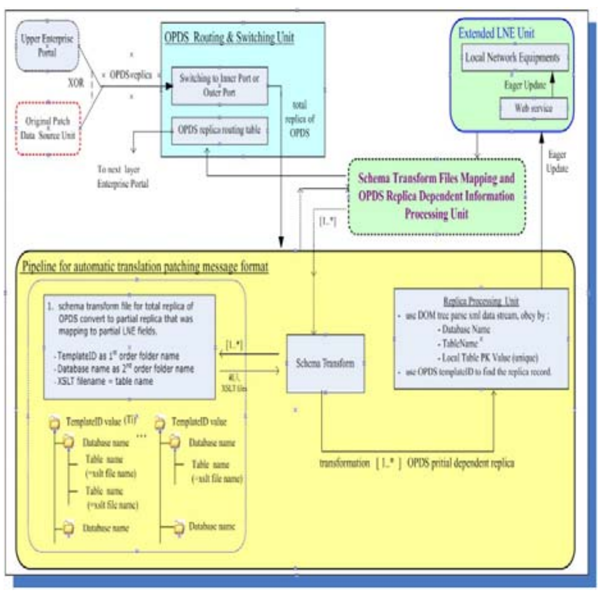

Figure 6. pipeline schema conversion

3.5 System Sequential diagram

In Figure 7 describe the UML sequential diagram about the concept model of system object.

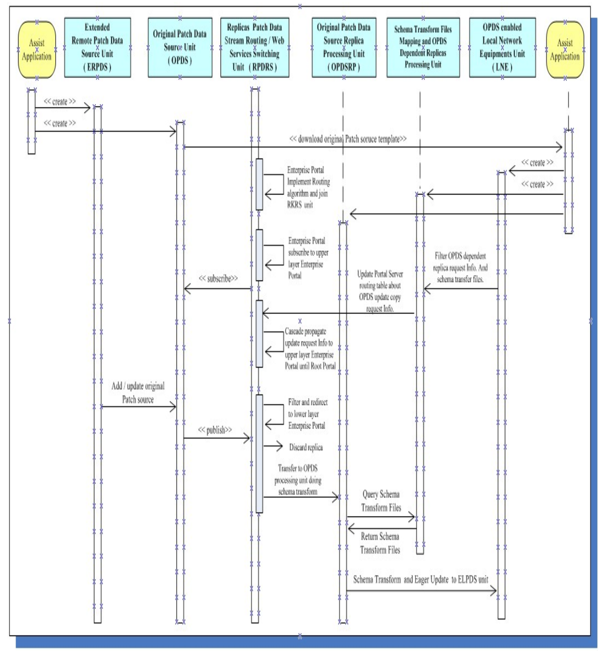

Figure 7. system object sequential diagram

4 Ripple propagation OPDS routing algorithm

Routing algorithm was built into each EP for assisting the xml data stream route path choosing.

4.1 Basic principle

The routing table fields description include: 1.OPDS dependency replica updates request information always maintained by EP 2.Information Included UUID value, PK value , Hop count, web service client etc. need to log in portal routing table from lower hierarchy enterprise request issues. 3. Portal's replica update routing information can come from intranet and extranet enterprise. 4. When a OPDS update replica publish to Portal, it will discard, send to processing unit, or reroute to lower portal depend on routing table check results. 5. For each route request from lower hierarchy portal will add a HC value to routing table's Hop Count value field.

Table 1. Routing table field's definition

Routing Table (RT) fields	Description
action state flag (ASF)	denote when add/delete OKMS dependent replicas in enterprise LKS
OKMS template ID (UUID)	OKMS template identify defined by OKMS site
OKMS replica Primary key value(PK)	instance of specific OKMS template
local connect / remote connect (LC / RC)	dependency replica update request issued by local enterprise or lower level enterprise portal
Destination Portal ID (DPID)	original enterprise Identify that issues the update request
Hop Count value (HC)	the distance between the enterprise contained the newest routing information and request enterprise
WSC_port	call next stop portal web service

4.2 Routing algorithms

Apply the rules to each route request. The ASF(Action State Flag) is sent from lower layer EP by portal agent and web services.

```
If (ASF==add) {
  // (rule.2-1) : ASF == add
    If (route request info.(UUID、PK、DPTD) sent from lower portal does not exist){
      //(rule.2-1-1)
      add this information into RT and upper cascade propagate this add event registry}
    Else If (route request info (WSC_port) identical){
        //(rule. 2-1-2-a)
          replace with the new route request info. and cascade propagate to upper portal when
          TTL trigger.}
    Else If (route request info (WSC_port) not identical){
        //(rule. 2-1-2-b)
        If (new HC value < RT's HC){
         //(rule. 2-1-2-b-I)
          replace with the new route request info. and cascade call back to lower hierarchy portal
          to disable the route with the same DPID when its WSC_port <> null }
        Else {
         //(rule. 2-1-2-b-II)
          Local portal do nothing then cascade call back to lower hierarchy portal to disable the
          route with the same DPID when its WSC_port <> null }

  }
Else {
// (rule.2-2) : ASF == delete
   If (route request info does not exist in RT){
    //(rule. 2-2-1)
    Local portal do nothing }
   Else {
    //(rule. 2-2-2)
     Find and delete the record in RT, and upper cascade propagate delete event registry }
 }
 Recursive()
```

Figure 8. DHWS routing algorithms

The following; Figure 9; is an demonstration for using above algorithm to adjust the P1 portal routing table

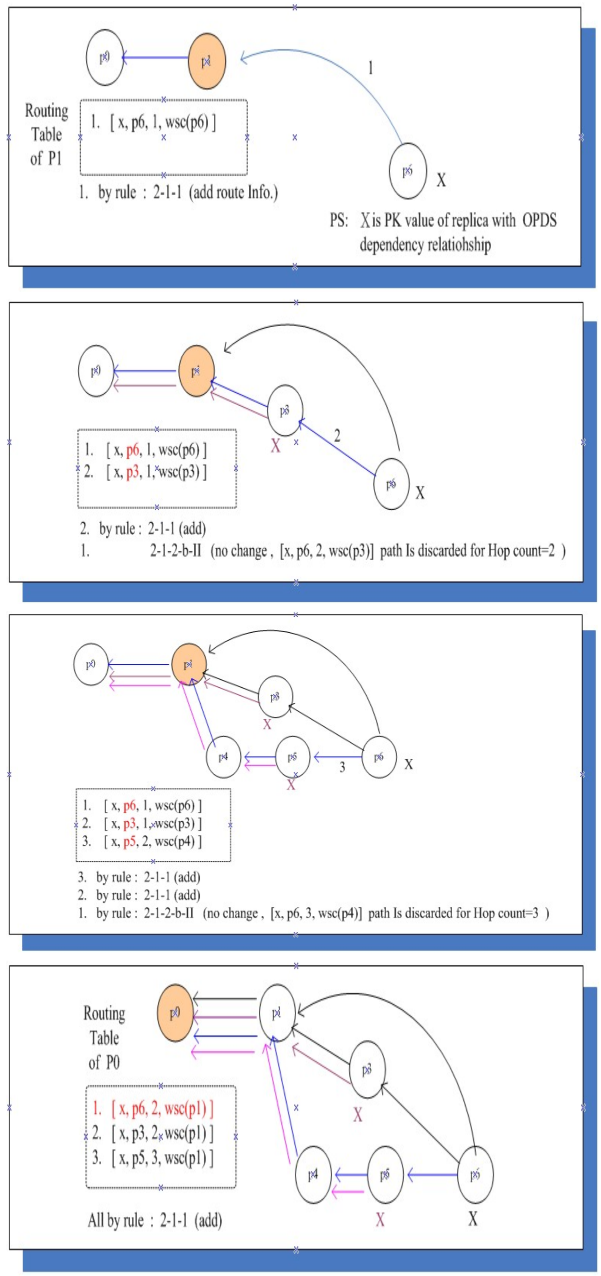

Figure 9. demonstration for routing algorithm

5. THGPCM model

5.1 Motivation

This section aims at deriving a mathematical model for the THGPCM forwarding plane to enable performance evaluation and optimal solution of the distributed routing data path. The analysis and modeling in fact leads to the well known shortest path optimization problem that has been extensively studied in the literature. Consequently, the paper will limit the analytical work to the derivation of the model and the mathematical expressions to minimize. The choice of an optimization criterion or performance metric determines the analytical model and the techniques that will be selected for optimization.

For simplicity, the paper focuses only on web service loss performance since adding an end-to-end constraint does not change the model. The objective is to minimize web service loss rate by selecting the appropriate set of FPC to form the routing data path. As described in Sect. 3.3 the distributed EP topology is composed of multiple interconnected EPs located in different Forwarding Portal Clusters. To enable load sharing over the THGPCM, the same EP function can exist in several Forwarding Portal Clusters. For a given EP, the next EP (or neighbor EP) may be located in the same Forwarding Cluster or in a different Forwarding Cluster. The distributed data path should take the shortest path, to minimize web service loss from the ingress to the egress EP along the logical distributed EP topology. It is more logical to forward web services from a given EPi to a next EPj located in the same Forwarding Cluster rather than send them to the EPj (with the same functionality) located in a different Forwarding Cluster. This is due to the web service loss that affects traffic performance between clusters (see Sect. 5.4, performance evaluation). Moving update patch replica from an EP to other EPs located in other clusters should be minimized. The ERP (Enterprise Root Portal), root of EP topology, is responsible for calculating and creating the THGPCM data path by configuring the EP web service forwarding tables involved in the data path web service processing. The objective for the ERP is to determine the most efficient path with respect to a given constraint (web service loss rate) to satisfy the requirements of multimedia traffic. This is known as the constraint-based path selection problem.

The ERP (Enterprise Root Portal) can get optimize web service routing table according by receiving dependency replica updating request from low layer EPs. Then ERP configure each EP web service forwarding table based on a shortest path algorithm to create the appropriate distributed data path.

5.2 Problem formulation

For a given data path, a set of EPs are needed to perform web service processing functions from an ingress EP to an egress EP, and these EPs can be physically distributed and can be duplicated. Hence, several paths are possible between ingress and egress EP nodes since multiple duplicated EPs can perform the distributed data path.

For a given data path, DP is used to denote a vector of EPj involved in the data path formation:

$$\text{DP} = (EP_j) \quad \text{for } 1 \le j \le F. \qquad (1)$$

that is also equivalent to a vector of sets A involved in the data path formation:

$$\text{DP} = (A_j) \quad \text{for } 1 \le j \le F. \qquad (2)$$

Figure 10 shows an example of a given FPC data path DP in enterprise intranet and extranet environment.

Symbol description： Enterprise is denoted by E. Routing path is denoted by r. flow path is denoted by c and b. EP encountered OPDS dependency patch replica flow will generate three kinds of message flow path, as follows.

Apply in enterprise extranet data path, like formula (3).

E5→b3→r1→r3→{c7|c8}→{E7|E8}→{b6|b

9}→r1→r3
(3)

For redirect updating replica to other hierarchy enterprise, flow path like formula (4).

E5→b3→r1→r2→{c1|c2|c3|c4}→{E1| E2| E3| E4 }→{b1|b2|b4|b7}→r1→r2
(4)

For synchronize trigger multiple EP horizontal patch replica consistency, the feedback intranet OPDS updating copy must be built in, like formula (5)

E8→b9→r1→{{r2 → same above (a)} | {r3 → same above (b)}}
(5)

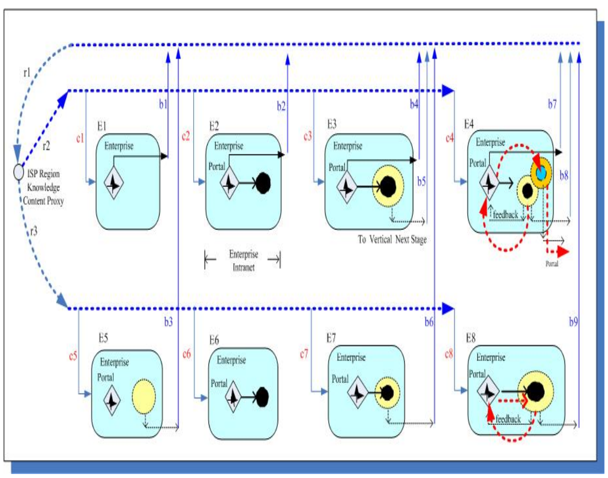

Figure 10. A graph model of Forwarding Portal Clusters (FPCs)

6 Conclusion and future work

This paper presents the design, implementation of an extensible, scalable, web services ripple propagate routable base, transnational hierarchical global patch consistent model. This model provide adaptive adjustable update routing policies, update reserve priority, and value added update services based on usage mining results of patch replica with specific OPDS dependency relationships. It's mainly contribution as follows. 1. The OPDS dependency patch replica usage mining rate is a function of dependency replica operation response time and turnaround time. 2. The updating route policy scheduling rate is a function of usage mining and reservation priority. The production rate of activity represented as a time variable. 3. Different value added update service schema was also analyzed. For example, active updating pushed by real time, periodic, reservation, and queuing with priority / degradation / preemption etc. With these resolution models, optimization of the correct copy inputs and switching time of each activity in a recursive routing are calculated to obtain the minimum total cost of the project. The result can reduce the global patch service cost of transnational enterprise network equipments and minimum the turnaround time of patch service delay.

References

[1] R. Ladin, B. Liskov, L. Shrira, and S. Ghemawat, Providing High Availability using Lazy Replication, *ACM Transactions on Computer Systems*, Vol. 10, No.4, 1992, pp.256-278.

[2] M. Patiño-Martinez, R. Jimenez-Peris, B. Kemme, G. Alonso, Consistent Database Replication at the Middleware Level, *ACM Transactions on Computer Systems (TOCS)*, Vol. 23, No.4, 2005, pp.1-49.

[3] Thanasis Loukopoulos, Ishfaq Ahmad, Static and Adaptive Data Replication Algorithms for Fast Information Access in Large Distributed Systems, *20th IEEE International Conference on Distributed Computing Systems (ICDCS'00)*, 2000, pp. 385-397.

[4] Yung Bok Kim and Soon Woo Lee, Performance Evaluation of Mobile Agents for

Knowledge-Based Web Information Services, *KES-AMSTA*, Lecture Notes in Computer Science, Vol. 4496, Springer, 2007, pp. 209-218.

[5] Ruey-Kei Chiu and Kuo-Chin Tsai and Chi-Ming Chang and S. C. Lenny Koh and Kuan-Chih Lin, The implementation of an agile information delivery system in building service-oriented e-healthcare network, *International Journal of Enterprise Network Management*, Vol. 1, Inderscience Publishers, March 14 2007, pp. 283-298.

[6] Ákos Hajnal and David Isern, Knowledge Driven Architecture for Home Care, *CEEMAS*, Lecture Notes in Computer Science, Vol. 4696, Springer, 2007, pp. 173-182.

[7] Aurora Vizcaíno and Juan Pablo Soto and Javier Portillo-Rodríguez and Mario Piattini, A Multi-agent Model to Develop Knowledge Management Systems, *HICSS*, IEEE Computer Society, 2007, pp. 203-209.

[8] Mahmoud Brahimi and Mahmoud Boufaïda and Lionel Seinturier, Integrating Web Services within Cooperative Multi Agent Architecture, *AICT/ICIW, IEEE Computer Society*, 2006, pp. 197-206.

[9] Nicolas Singer and Jean-Marie Pecatte and Sylvie Trouilhet, Combining Web Services and Multi-Agent Technology to Increase Web Cooperative Capacities, *International Conference on Internet and Web Applications and Services (ICIW'07)*, Mauritius, IEEE, May 2007, pp.367-375.

[10] Oh Byung Kwon, Multi-agent system approach to context-aware coordinated web services under general market mechanism, *Decision Support Systems*, Vol.41, No2, 2006, pp. 380-399.

[11] Anna Maria Jankowska and Karl Kurbel and Denny Schreber, An architecture for agent-based mobile Supply Chain Event Management, *International Journal of Mobile Communications*, Vol. 5, Inderscience Publishers, February 11 2007, pp. 243-258.

[12] D. Serrano, M. Patiño-Martinez, R. Jimenez-Peris, An Autonomic Approach for Replication of Internet-based Services, *IEEE Symposiumon Reliable Distributed Systems (SRDS)*, 2008, pp.244-255.

[13] Hakan Tozan, Ozalp Vayvay, Fuzzy Forecasting Applications on Supply Chains, *WSEAS Transaction on Systems*, Vol. 7, Issue 5, 2008, pp. 600-609.

[14] Chen Zhao, Fei Gao, The Business Process Model for IT Service Management, *WSEAS Transaction on Systems*, Vol. 7, Issue 12, 2008, pp. 1494-1503.

[15] Ákos Hajnal and David Isern, A Sampling-based Method for Dynamic Scheduling in Distributed Data Mining Environment, *WSEAS Transaction on Computers*, Vol. 8, Issue 1, 2009, pp. 63-72.

[16] Y. Aimirk A. Peterson, D. Shaw m, Seamlessly selecting the best copy from internet-wide replicated web servers, *Distributed Computing*, Lecture Notes in Computer Science, Vol. 1499, Springer, 2006, pp. 23-35.

[17] J. Sloten, A. Pras, M. Sinderen, "On the Standardisation of Web service management operations", *Proc. 10th Open European Summer School (EUNICE 2004) and IFIP WG 6.3 Workshop* (June 2004), pp. 143-150.

[18] T. Drevers, R. Meent, A. Pras, "Prototyping Web Services based Network Monitoring", *Proc. 10th Open European Summer School (EUNICE 2004) and IFIP WG 6.3 Workshop,* 2004, pp. 135-142.

[19]. G. Pavlou, P. Flegkas, S. Gouveris, A. Liotta, "On Management Technologies and the Potential of Web Services", *IEEE Communication Magazine(My* 2004), pp. 58-66.

[20]. J. Schonwalder, A. Pras, J.P. Martin-Flatin, "On the Future of Internet Management Technologies", *IEEE Communications Magazine{OctobeT* 2003), pp. 90-97.

[21]. W3C, "SOAP Version 1.2 Part 0: Primer", *W3C Recommendation,* April 2007.

[22]. W3C, "SOAP Version 1.2 Part 1: Message Framework", *W3C Recommendation,* 2007.

[23]. W3C, "SOAP Version 1.2 Part 2: Adjuncts", *W3C Recommendation,* 2007.

[24]. W3C, "Web Services Description Language (WSDL) Version 2.0 Part 0: Primer", *W3C Proposed Recommendation,* May 2007.

[25]. W3C, "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language", *W3C Proposed Recommendation,* May 2007 .
[26]. W3C, "Web Services Description Language (WSDL) Version 2.0 Part 2: Message Exchange Patterns", *W3C Working Draft,* March 2004 .
[27]. W3C, "Web Services Description Language (WSDL) Version 2.0 Part 3: Bindings", *W3C Working Draft,* August 2004.