

Approach of service recovery decision-making based on Bellman dynamic programming

He Lei Ren Jiangchun Wang Zhiying

(School of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: Based on service-oriented architecture (SOA), a Bellman-dynamic-programming-based approach of service recovery decision-making is proposed to make valid recovery decisions. Both the attribute and the process of services in the controllable distributed information system are analyzed as the preparatory work. Using the idea of service composition as a reference, the approach translates the recovery decision-making into a planning problem regarding artificial intelligence (AI) through two steps. The first is the self-organization based on a logical view of the network, and the second is the definition of evaluation standards. Applying Bellman dynamic programming to solve the planning problem, the approach offers timely emergency response and optimal recovery source selection, meeting multiple QoS (quality of service) requirements. Experimental results demonstrate the rationality and optimality of the approach, and the theoretical analysis of its computational complexity and the comparison with conventional methods exhibit its high efficiency.

Key words: service recovery decision-making; Bellman dynamic programming; quality of service (QoS); service-oriented architecture(SOA)

With the rapid development of network and distributed computing, web services and service composition have become typical application patterns in the open cooperative network environment^[1]. Giving full play to the advantages of services, such as public services, accessibility and interoperability^[2]. The service-oriented architecture (SOA) has already been the mainstream architecture in organizing a new generation of web services^[3~4]. Information technology has been applied to many crucial areas of the national economy and security on a large scale. Therefore, how to guarantee the service stability and business continuity of information systems under catastrophic situations has become an urgent issue in the information security domain.

In current researches on disaster recovery regarding information systems, recovery decision-making always adopt the methods of recovery from local or remote data backups^[5~6]. Such method does not make full use of web services' characteristics, and it is not based on global network resources. In this paper, we propose a concept of service recovery based on the SOA. Service recovery includes emergency response to client requirements and transparent recovery to destroyed services, by planning network service resources as a whole. It also selects correct recovery technologies, including dynamic migration, reconstruction and degradation, in terms

of the feature of destroyed services. Service recovery can guarantee the abilities of emergency responses and disaster recovery in information systems efficiently and reliably.

Recovery decision-making is the key step in the process of service recovery. It includes the selection of the response path, the recovery source set and the method. In this paper, the technology of service composition is used to support service recovery. The recovery source set is made up of related sub-services. We apply dynamic programming theory to recovery decision-making in controllable distributed information systems, and translate the problem into an AI planning problem correctly. Bellman dynamic programming^[7] is used to map out the recovery strategy by dividing the problem into a set of smaller ones. Utility function is defined as the evaluation standard considering the quality of service (QoS)^[8] parameters of each service component. Our method successfully guarantees the business continuity of information systems by quantitative decision-making, reflecting the dynamics and the randomness of distributed information services. And it is more efficient and dynamic than the conventional methods in Refs. [9 – 10].

1 Dynamic Programming Theory

Bellman's dynamic programming^[7], one of the best solutions to multi-stage planning problems, can achieve global optimal results. In applications, when the state variable has the feature of no after-effects, optimization theory comes down to a quantitative recursion which continuously advances the multi-stage planning process.

The enterprise application is the integration of a set of small granularity services in the SOA environment, which conforms to a certain logical sequence^[11]. According to the sequence, a service process can be divided into some related stages. In the same way, the process of choosing recovery source sets can be divided according to the types of sub-services in the destroyed one, regarding the level of the service composition as the system state. Thereby, the problem of service recovery decision-making has been translated into a multi-stage decision-making problem.

To solve a multi-stage decision-making problem, we have to make a decision at every stage. The former choice has a direct effect on the latter one, consequently, influencing the whole strategy. Decisions from every stage form a decision-making sequence, which is entitled as a "strategy", while the decision of every stage is called a "choice". Because of the various choices in every stage, there are many strategies to select from. We choose the best strategy according to an optimal utility function. In this paper, we define the utility function considering QoS parameters, and construct the recursive relationship between the strategy and the choices according to the dynamic programming theory, in order to ob-

Received 2008-04-15.

Biographies: He Lei (1984—), female, graduate; Wang Zhiying (corresponding author), male, doctor, professor, zywang@nudt.edu.cn.

Citation: He Lei, Ren Jiangchun, Wang Zhiying. Approach of service recovery decision-making based on Bellman dynamic programming[J]. Journal of Southeast University (English Edition), 2008, 24(3): 377 – 380.

tain the global optimal strategy.

2 Problem Analysis

2.1 Description of service process

By analyzing the composite type and mode of web services, we infer that a service process can always be translated into the form of N kinds of sub-services processing in turn. Crucial service path is used to represent the service process. To facilitate interpretation, we give the following definitions^[12].

Definition 1 Meta-service is the minimum division unit of services, which is the smallest independent element with a special function.

Definition 2 Component service consists of one or more meta-services according to combinational logic, and provides a special function. It is a component of more complex services.

Definition 3 Composition service consists of one or more component services according to combinational logic, providing a value-added service.

Definition 4 Service type is the function provided by a service.

Definition 5 Service mode is the cooperative form of component services, including sequential mode; i. e. components respond to a requirement in turn, and collaborative mode; i. e. components respond mutually.

Definition 6 Recovery source set is the set of component services, which composes a composition service serving as the service to be recovered.

Definition 7 Crucial service path is the processing sequence of component services in a composition service.

For a given service S , its crucial service path is $S_1 \rightarrow S_2 \rightarrow S_3, S_4 \rightarrow S_5$. S_1 to S_5 represent types of sub-services, while “,” and “ \rightarrow ” are symbols of the collaborative and sequential models, respectively.

2.2 Mapping from decision-making to planning

The problem of service recovery decision-making is translated into a multi-stage decision-making problem, using a crucial service path to describe the service to be recovered. To facilitate the formalized description and agent realization, we map the decision-making to a planning in the AI field. Component-services are actions in the planning, and the completion level of the service to be recovered is the state. Thus, the recovery of a complex service is translated into the combination of a set of distributed service resources. The function of service recovery decision-making includes arranging web resources and selecting the right recovery source set and recovery technology.

Step 1 Self-organization on logical view of the network

The partition of stages is achieved along the crucial service path. Divide the planning according to the stages and achieve the available service-node connection graph (ASCG), in terms of the network topology and the service nodes connections of the studied distributed information system, as illustrated in Fig. 1. R represents the destroyed service node. When a client applies for R -service, which is destroyed, the requirement will be delivered to the recovery source set to be processed cooperatively. And the response will be returned to the client by R in the end. Thus, the cli-

ent receives the response in time, even though part of the system has been destroyed. Meanwhile, service recovery to R is realized transparently to the client. Nodes in Fig. 1 represent service-nodes in the information system. Edges are the connections between nodes. Weights can be customized according to user requirements. In an ASCG, the destroyed node is the source node, and is also the destination node. “ \rightarrow ” indicates the existence of a connection, and the direction conforms to the processing sequence of sub-services.

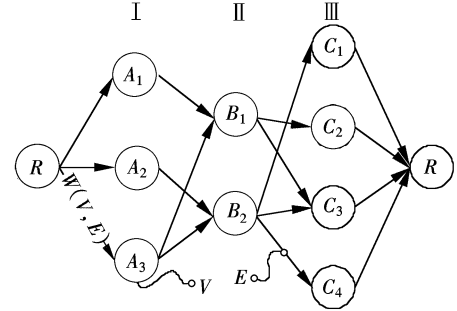


Fig. 1 An ASCG

An ASCG is constructed as follows: 1) If there are one or more available service nodes in the service pool for every required service type, construct the ASCG according to connections between nodes; 2) If there is no service node of some required service type, delete the type from the requirement and employ the technology of service degradation, and then process as in 1); 3) There is no available service node of some required service type, but not as in the situation of 2). Compare the priority of the current requirement with the appropriator's. If the current requirement has a higher priority, consider the service node to be available and process as in 1). Else, wait for adequate available sources.

If there is a collaborative mode in the crucial service path, the collaborative services are considered to be a complex component service, shown in Fig. 2. At that time, full connection is required among the collaborative nodes and the service nodes of the former and latter stages.

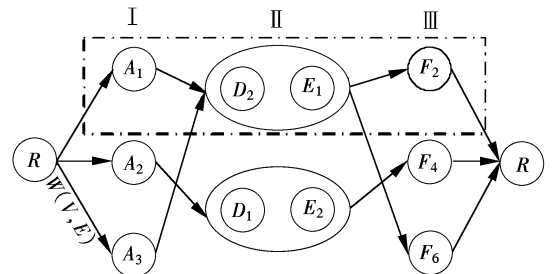


Fig. 2 An ASCG with collaborative services

Step 2 Definition of evaluation standard

For a given service, some non-functional properties, such as price, duration, reliability, availability, reputation and so on, are selected as reference standards of the QoS. The computing cost of a service is described as follows: $T_c = T$ (price, duration, reliability, availability, reputation, ...)

Definition 8 Stage cost is the cost of a certain choice at a given stage. In this paper, the weighted sum of computing and transfer time is defined to represent the stage cost; i. e. $w_k = w_1 T_c + w_2 T_t$. In applications, we can employ other QoS

parameters to define the stage cost according to the actual requirements.

Definition 9 Utility function is represented as the sum of stage costs, i. e. $\sum_{k=1}^n w_k$. It is the evaluation standard of recovery.

Definition 10 Optimal utility function is the minimum of the utility function.

For a given service, constructing the ASCG according to step 1, defining the valid utility function as step 2, then the recovery decision-making is transferred to a planning. The future study is the automatic solution to the planning.

3 Description of Planning Algorithm

Bellman dynamic programming is used to solve the problem above. The global optimal strategy of service recovery can be achieved through back-forth recursion without roll-back. At any stage of planning, there will not be inappropriate decisions. The details are described as follows:

- 1) Select correlative service nodes which provide the needed types of service.
- 2) Form the ASCG to the requirements, in terms of the network topology and the crucial service path.
- 3) Define the utility function. According to the QoS parameters provided by the UDDI Registration Centre, calculate stage costs, and transform the ASCG achieved in 2) into a weighted ASCG.
- 4) Check the weighted ASCG's connection, i. e. whether the ASCG is from R to R . If so, go to 5). If not, adopt the technology of service degradation and continue.
- 5) Use Bellman dynamic programming to achieve the optimal response path and the recovery resource set. Select the technology of dynamic migration or reconstruction, in terms of the scale of the problem.

3.1 Application of Bellman dynamic programming

The service process is divided into a series of stages when employing Bellman dynamic programming to solve the problem. The service node represents the state, and the choice of edges represents sub-strategies. When the sub-strategy is made, the system state is transformed into another one, with a certain stage cost.

The settlement of strategy is presented by the global strategy and the rear strategy. The sequence of choices from initial-state S to end-state R forms the global strategy, and the sequence of choices from state V_k to end-state R forms the rear strategy, i. e., $p_k(V_k) = \{E_k, E_{k+1}, \dots, E_n\}$. The optimal utility function is defined as the optimum of the sum of the stage costs. $d_k(v_k) = \text{opt}_{p_i \in P_i} \left(\sum_{j=k}^N W_j \right)$, P_k is the set of all possible rear strategies, $\sum_{j=k}^N W_j$ is the cost of a certain rear strategy, and $d_k(v_k)$ is the best one. It is generalized as

$$d_k(V_k) = \text{opt}_{E_i \in D_i(V_k)} \{W_k + d_{k+1}(V_{k+1})\}$$

$$d_{N+1}(V_{N+1}) = 0$$

$D_k(V_k)$ is the set of choices at the state V_k . Working out the generalized optimal utility function, we can achieve the optimal global strategy for the planning.

3.2 Pseudo advanced language description

The above recovery decision-making algorithm is described in pseudo advanced language as follows. The ASCG and corresponding weights have already been achieved, and they serve as inputs of the function. S and R represent the source and destination nodes of the ASCG, respectively. After the Bellman dynamic programming is recursively applied to solve the graph, the optimal path from S to R and its cost estimates are provided as outputs.

Function Bellman-decision (G, W, S, R) returns D and d .

Inputs: G is an ASCG, $G = G(V, E)$; W is the weight, $W = (w_{ij})$; S is the source of G ; R is the destination of G .

Outputs: D is the optimal path from S to R ; d is the cost estimate of the optimal path from S to R .

Local variables: $D[v]$ is the v 's successors; $d[v]$ is the optimal-path estimate from v to R .

```

Initialize-single-source ( $G, S, R$ );
do for each edge  $(S, u) \in E[G]$ 
do  $D[u], d[u] < -\text{Bellman-decision}(G, W, u, R)$ 
if  $d[S] > d[u] + w(S, u)$ 
then  $d[S] < -d[u] + w(S, u)$ 
 $D[S] < -\{u, D[u]\}$ 
endif
endfor
return True
Function Initialize-single-source( $G, S, R$ )
for each vertex  $v \in V[G]$ 
do  $d[v] < -\infty$ 
 $D[v] < -\text{NIL}$ 
endfor
 $d[R] < -0$ 

```

4 Implementation and Analysis of Algorithm

We implement the above recovery decision-making algorithm with Java programming language on a Windows XP SP2 system platform. The approach gives emergency responses to user requirements, and chooses the optimal recovery set of service candidates and correct recovery technology. Its rationality and validity are guaranteed by the global optimality of Bellman dynamic programming.

The computational complexity of the approach is analyzed by a theoretical analysis method. And the scale of the given problem is described as follows. In a distributed information system, the destroyed service to be recovered averagely contains M types of sub-services, and the mean number of service nodes providing a certain type of function is N .

The space and time cost of the algorithm is mainly from two parts: One is the space and time spent on the process of self-organization on logical view of the network is $o(MN^2)$. The other part of the cost is also $o(MN^2)$, which is used to complete the Bellman dynamic programming. To sum up, the whole computational complexity is about $o(2MN^2)$. While the conventional approach of blind search needs about $o(N^M)$. Obviously, our approach to recovery decision-making is more efficient. Meanwhile, with the development of the Internet and web services, the scale of application integration increases correspondingly. The higher values of M and N will make the efficiency of our approach more significant.

We employ the technology of service recovery to recover a destroyed book-market-service in LAN as an example. A book-market-service consists of four parts such as directory, book-management, order-management and payment service, which represent four stages in the process of service recovery. Formula $w_k = w_1 T_c + w_2 T_t$ is used to describe the cost of the k -stage. T_c represents the computing time of a service nodes, and T_t is the time of transfers. The former time is determined by the service node itself, while the latter one depends on the speed and length of the link. w_1, w_2 are acquired with experience, indicating weights of the two kinds of time. Test scene: the capacity of a service pool is 50; the number of service types is 10; the priority of the requirements are 3; and $w_1 = 1, w_2 = 1$.

According to the algorithm in section 3, we solve the pro-

blem of recovery to book-market-service and obtain the correlative weighted ASCG, as illustrated in Fig. 3. The bold lines give the path of emergency response, and results of the algorithm are given in Tab. 1.

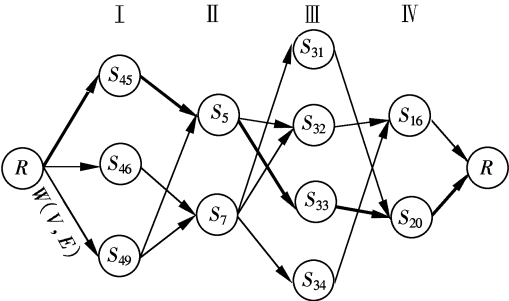


Fig. 3 Weighted ASCG of book-market-service

Tab. 1 Result of recovery decision-making to book-market-service

Requirement description		Result	
Crucial service path	Priority	Recovery method	Recovery source set
$S_9 - > S_1 - > S_6 - > S_4$	3 (highest)	Non-degradation; reconstruction	$S_{45}, S_5, S_{33}, S_{20}$

5 Conclusion

Recovery decision-making is the hard core of service recovery. The algorithm based on Bellman dynamic programming proposed in this paper solves the recovery decision-making problem successfully. It is conducive in guaranteeing the business continuity of distributed information systems, improving the ability of emergency response, and accelerating the transparent recovery from disasters. Due to space limitations, the auto-creation of evaluation standards to various requirements is planned as a future improvement.

References

[1] Ganesarajah D, Lupu E. Workflow-based composition of web-services: a business model or a programming paradigm[C]// *Proc EDOC'02*. Lausanne, Switzerland, 2002: 273 – 284.

[2] Wang H, Huang J Z, Qu Y, et al. Web services: problems and future directions [J]. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2004(1): 309 – 320.

[3] Chen Lin. Research and application of service oriented architecture[D]. Shanghai: East China Normal University, 2004. (in Chinese)

[4] Papazoglou M P. Service-oriented computing: concepts, characteristics and directions [C]// *Proc WISE'03*. San Diego, USA, 2003: 3 – 12.

[5] Toigo J W. *Disaster recovery planning: for computers and communication resources* [M]. New York: Prentice Hall PTR, 2001.

[6] Hvasshovd S O. *Recovery in parallel database systems* [M]. 2nd ed. Wiesbaden: Vieweg, 1999.

[7] Bellman R E. *Dynamic programming* [M]. New York: Courier Dover, 2003.

[8] Ren K, Chen J, Chen T, et al. Grid-based semantic web service discovery model with QoS constraints [C]// *Proc SKG'07*. Xi'an, China, 2007: 479 – 482.

[9] Liu J, Cui J, Gu N. Composing web services dynamically and semantically[C]// *Proc of IEEE Conference on E-Commerce Technology for Dynamic E-Business*. Beijing, China, 2004: 234 – 241.

[10] Cheung W K, Liu J, Tsang K H, et al. Towards autonomous service composition in a grid environment [C]// *Proc ICWS'04*. San Diego, USA, 2004: 550 – 557.

[11] Rao J. Semantic Web service composition via logic-based program synthesis[D]. Norwegian: Norwegian University of Science and Technology, 2004.

[12] Foster I, Kesselman C, Tuecke S. The anatomy of the grid: enabling scalable virtual organizations [C]// *Proc Euro-Par'01*. Manchester, UK, 2001: 1 – 4.

基于贝尔曼动态规划的服务恢复决策方法

何 蕾 任江春 王志英

(国防科学技术大学计算机学院,长沙 410073)

摘要: 基于面向服务的架构(SOA), 针对可控分布式信息系统中服务的特点并借鉴服务组合思想, 详细分析了信息系统的典型服务过程, 提出一种基于贝尔曼动态规划的服务恢复决策算法. 通过逻辑层网络自组织和评价标准制定 2 个步骤, 将服务恢复过程中恢复决策问题转化为人工智能领域的规划问题, 并对此规划问题使用贝尔曼动态规划算法, 解决了确保服务质量(QoS)的前提下, 最优服务恢复方案制定与应急响应路径选择问题. 仿真实验验证了方法的最优性与合理性, 算法复杂性理论分析及与传统方法的性能对比均表明了该方法的高效性.

关键词: 服务恢复决策; 贝尔曼动态规划; 服务质量; 面向服务的架构

中图分类号: TP309