第 32 卷 第 23 期 计 算 机 工 程 2006 年 12 月
Vol.32 No.23 Computer Engineering December 2006
·博士论文· 文章编号：1000—3428(2006)23—0030—04 文献标识码：A 中图分类号：TP393

基于服务生成图的 Web 服务工作流可用性研究

胡春华[1,2]，吴 敏[1]，刘国平[1,3]，王四春[2]
(1. 中南大学信息科学与工程学院，长沙 410083；2. 湖南商学院计算机与电子工程系，长沙 410205；
3. School of Electronics, University of Glamorgan, Pontypridd CF37 1DL)

摘 要：针对 Web 服务工作流随着服务的动态产生和消失，导致工作流的可用性受影响的现象，提出了一种基于服务虚拟平台的服务工作流可用模型——High-Workflow，论述了可用模型的结构体系、组织协议。该模型有效地屏蔽了组成工作流的 Web 服务物理上的差异，通过 Web 服务的复制与备份提高了工作流的可用性，在理论上分析了模型的可用度，通过实例验证了该方法的有效性。
关键词：服务网络；可用性；工作流；备份服务

Research on Availability of Web Service Workflow Based on Service Spanning Graph

HU Chunhua[1,2], WU Min[1], LIU Guoping[1,3], WANG Sichun[2]
(1. College of Information Science and Engineering, Central South University, Changsha 410083; 2. Department of Computer & Electronic Engineering, Hunan Business College, Changsha 410205; 3. School of Electronics, University of Glamorgan, Pontypridd CF37 1DL)

【Abstract】 Aimed to the problem that the availability of Web service workflow is influenced for the dynamic bringer and disappearance of service in the Internet, this paper proposes a method of available service workflow based on service virtual system, and analyzes the method's structure architecture and organized protocols. It can effectively shield the physical difference of Web service and improve the availability of workflow by copying and backuping Web service in Internet. In the end, the validity is tested by the theory analyses and a real application.
【Key words】 Service network; Usability; Workflow; Backup service

1 概述

Web 服务正在成为一种基于标准技术的计算平台，它支持构建松散耦合的广域分布式系统，在工作流领域得到了很好的应用，基于 Web 服务的工作流是通过服务之间定义的接口，使工作流中的每一个服务成为另外一个服务的“上游”或者“下游”，由于 Web 服务具有通用的协议标准，从而使得基于 Web 服务的工作流获得了广泛的应用。

目前，基于Web服务的工作流可用性是一个重要的问题，主要原因在于工作流的组成是由多个服务的协作才完成一个完整的工作流，工作流中的任何一个执行点发生失效就会使整个工作流受到影响。目前，纯粹单个服务的可用性研究也存在许多挑战和问题。首先还没有一个全球性的服务组织网络，网络提供的服务往往是动态生成，而又随时消失的，导致服务网络中虽然存在着数量非常巨大的服务供给集合，但这些服务集合对于用户来说非常难以得到应用，其主要原因在于，服务提供者很难在全球范围内推广。对用户来说，通过各种搜索平台得到的服务往往是过时的信息，实际的服务早已经消失，导致用户感觉网络服务的不可用，但实际上，尽管服务网络上的服务失效非常频繁，但对整个服务网格来说，其平均无故障时间很短[1,2]。

局域网中服务器的可用性机制已经有很多研究，但这些研究都还不能扩展到全球以服务为基本单元的可用性研究中，更未能够运用到分布式的协作工作流中这样一种多个服务的可用性问题上。总之，服务工作流可用性的研究主要包括：(1)如何在这样的一个动态的服务网络计算环境中，将服务有效地组织起来，并根据应用的需要，采用一定的服务备份机制，保障服务网络提供具有可用性需求的服务；(2)对于给出的某一工作流，如何计算其可用性，如何确定备份哪些服务、备份的数量，以保障工作流达到应用的需求。而要解决这 2 个问题的关键在于对广域网络中服务资源的有效组织，在对服务资源有效组织的基础上就可以对组成工作流的各个服务进行可用性分析，对工作流的可执行路径进行分析，从而得到执行路径中的关键服务、关键路径，并采用一定的措施以提高工作流的可用性，基于这种思想将组成工作流的服务用服务生成图来表示，建立了一个服务的虚拟平台，在此平台上进行工作流可用性的管理，从而屏蔽了服务物理上的差异，使得模型能够适应高度动态的服务环境，提供有保障的工作流，这种基于虚拟平台的方法也正是当前面向服务的下一代网络所采用的方法[3,4]。

本文首次提出了广域互联网中服务工作流的可用性问题，并提出了用工作流服务生成图来支持工作流的可用性，工作流生成图依据接口关系将组成工作流的不同服务组织成不同的服务集合，每一个服务集合采用生成树的形式来组织与管理，生成树与生成树之间按照工作流的流程关系形成“生成图”，生成图代表了工作流的虚拟服务系统，由生成图来维

基金项目：国家杰出青年科学基金资助项目(60425310)；教育部青年教师奖基金资助项目(教人[2002]5 号)；湖南省自然科学基金资助项目
作者简介：胡春华(1973－)，男，博士生、讲师，主研方向：网格计算，工业以太网；吴 敏、刘国平，博士、教授、博导；王四春，博士、教授
收稿日期：2006-05-22 **E-mail：**huchunhua777@163.com

护工作流的可用性，由于提出的生成图是一种动态可适应的分布式组织系统，具有全球的服务组织能力与动态的可适应性，在这些基础上，再根据对工作流的可用性要求，确定生成图各服务组成的服务个数，从而使网络服务对应用(工作流)具有系统层面上的可用性。

2 基本术语

定义 1 工作流的可用性：对于某一个工作流应用来说工作流的可用性是指此工作流应用得到成功执行可能性的概率。

定义 2 服务间可替代关系：如果某个应用能在服务WS_i上满足性能要求，也能在服务WS_j上得到性能满足，则称服务WS_i可替代服务WS_j，记为$Sim(WS_i,WS_j)$。

定义 3 Web服务生成树(Web Services Span Tree, WSST)，是一棵生成树，用无权无向图表示，即G=(V,E)，其中V=n，是树中服务节点的个数；E=m，是树中边的个数；服务节点为提供同一类服务的节点，且对于每一个生成树$WSST_k$的任意 2 个服务(WS_i,WS_j)，都有$Sim(WS_i,WS_j)$。

定义 4 Web服务生成图(WSSP)，是依据工作流的流程关系由多棵生成树组合成的图，可以用G=(T,E)，其中T=WSST(Web服务生成树的集合)，是生成图中构成工作流的各个Web服务生成树；E=<T_i,T_j>，即生成图中二棵生成树之间接口的流程关系组成的边；T_i树中服务的“下游”是T_j，即T_i树、T_j树中服务接口形成工作流中的路径。

定义 5 工作流模式(Workflow model)，是指组成工作流的各个服务类型相互关系而形成的工作流图，如图 1 所示。

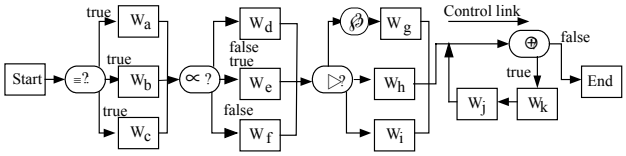

图 1 工作流模式

值得注意的是，工作流模式中的服务是指一种服务的类型，而不是具体的服务实例，属于此类型的服务实例均可以完成此类型所代表的活动，工作流模式可以描述为

WM=(MODELS, CL)

其中，(1)MODELS={W1,W1,...,WN}为服务类型的集合，W1,...,WN 表示工作流的 N 个参与的服务类型；(2)CL⊆MODELS×MODELS×φ为控制连接(Control Link)的集合。对于<Wi,Wj, φ>∈CL，控制结构运算符φ定义为从Wi到Wj的控制转换关系。

基本服务通过若干控制结构可组合成服务工作流，服务工作流本身就是一个复合服务。工作流使用的控制结构运算符依据 BPEL 的定义，BPEL 结构化活动共有 5 个，分别是：

(1)顺序控制(sequence)：包含一组顺序执行的活动，用→表示，如$(W_1→W_2)$。表示服务W_1和W_2的顺序执行；

(2)选择控制(switch)：根据条件表达式定义的判断条件进行不同服务的路由选择，用 ∝ 表示，如$(\propto ?W_1,W_2,W_3)$表示根据条件 ∝ 选择W_1、W_2和W_3中的一个服务；

(3)循环控制(while)：根据条件为真时循环执行服务，用⊕表示，如⊕ (W_1)表示⊕为真循环执行W_1；

(4)并行控制(flow)：定义了一组并行执行的服务，当这些服务都执行完成时，该并行服务就完成，用 ≡ 表示，如$(\equiv :W_1,W_2,W_3)$表示W_1,W_2,W_3都执行完毕则并行服务执行；

(5)事件选择活动(pick)：该活动包含一组事件，每个事件都和某个服务相关。一旦某个事件到达，和该事件关联的服务就会被执行，当该服务执行完后pick服务也完成了，用$\wp$表示，如$\wp(W_1)$表示依据事件$\wp$激活W_1中的对应服务执行。

用户还可以扩展控制结构，如 ⊳ 表示优先选择$(W_1 ⊳ W_2)$，表示在执行过程中总是试图先执行W_1,若W_1执行失败，则执行W_2。

一个服务工作流控制结构表达式的例子：$(\equiv :W_a,W_b,W_c)→(\propto ?W_d,W_e,W_f)→((W_g,\wp(W_h) ⊳ W_i)→(\oplus[W_k→W_j])$。其对应的结构如图 1 所示，用实线表示控制流，服务类W_a-W_k对应于实际服务中的一棵生成树。

3 可用性模型

3.1 可用性模型结构

服务工作流的可用性结构模型 High-workflow 如图 2 所示。系统主要由 2 个部分组成：服务注册系统(Service Register System, SRS)和服务工作流生成图(Service Workflow Spanning Graph, SWSG)。

(1)服务注册系统(SRS)：它主要起到索引服务作用，在文献[5,6]中提出了一种与 DNS 类似的分布式 Web 服务注册系统，在其基础上略做改进，主要的区别是：在文献[7,8]中主要是利用与 DNS 类似的服务注册系统，每一个服务提供者直接向 SRS 注册。在本文中采用了一种生成图的方式来组织工作流所对应的服务，在服务的组织中，并不让每一个服务供者直接向 SRS 注册，由于服务的动态产生与消失现象比 DNS 系统复杂，因此直接运用与 DNS 类似的注册方法使服务的动态性信息的维护性不高，系统的负载压力大，在本文中对于一棵生成树来说，选中其中一个作为令牌节点，由其来负责整棵树与 SRS 的服务信息维护，这样对于广域网中的一类服务只需要一个节点与 SRS 交互就可以了，大大减少了服务信息维护的负载压力。如果该令牌节点失效，在生成树中重新选择一个作为新的令牌节点。

(2)Web服务生成图：对于某类应用互联网中存在所有$Sim(WS_i, WS_j)$的服务形成一棵生成树，对于每一棵树有一个令牌节点[7]，它负责树中所有信息的维护，它与SRS间存在一个定时的交互动作TTL(Time To Live)，目的是让DHT知道树还“活着”，它由令牌节点在TTL交互时将树内信息报告给DHT。依据工作流模式，将服务间的“上下游”对应的生成树之间关联起来，形成服务生成图，如图 2 所示。一个工作流的可用性则可以由工作流模式和服务生成图计算得到，而采用生成图的方式来组织服务有如下几个特点使得系统计算可用性非常方便。

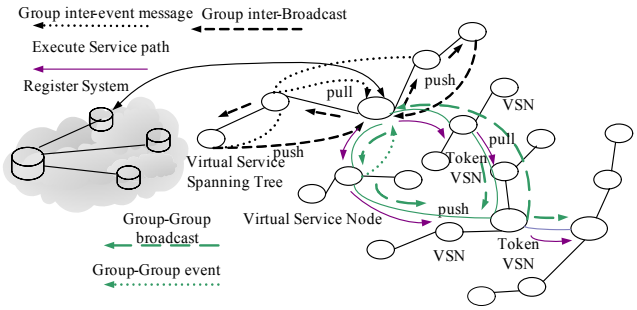

图 2 Web服务工作流可用性模型

采有 Web 服务生成图的原因如下：

(1)采用一棵树的形式来保持信息的动态信息维护，有效地减少了每个服务直接向 DHT 交互所带来的负载压力，这种组织结构虽然内部服务动态性很大，但 WSST 作为一个整体，

仍然对外提供服务，系统的可用性大大提高了；

(2)有效的服务组织：互联网中服务的动态产生与失效非常频繁，用户查找服务非常困难，在将某类应用的所有可替代服务组织成一树生成树，当互联网存在此类服务时则此生成树存在，生成树中节点的个数表示提供此类服务个数的多少；在组织成生成树后，用户查找服务非常方便与快速，它只要先向 DHT 查询生成树的地址，DHT 返回生成树的地址后，用户然后向生成树提交服务，请求得到服务；

(3)比照工作流模式：采用本文给出的可用性算法，很容易得到工作流中的关键服务、关键执行路径，从而可以有针对性地增加有效的服务提供者，而使整个系统的可用性得到提高，以往的系统中无法做到及时分析工作流的可用性问题，也无法准确有效地增加服务提供者的数量来提高系统的可用性。

本系统易于实施服务的可用性控制，对于一棵生成图来说，它是一种自我维护的、分布式的系统。下文给出了详细的维护方式。

对于一个服务生成图来说，存在的操作维护过程有：(1)WSST 的形成过程与生成图的形成过程，此类过程主要是为了使系统结构的正常成长与运行；(2)WSST 内的服务信息维护算法以及生成图的信息维护过程，它的主要目的是为了维护信息的及时性、准确性和可用性。

3.2 Web 服务生成树的生成算法

当一个服务产生后，是根据如下的生成算法产生一棵 Web 服务生成树。

算法 1 Web 服务生成树的构造算法

step1 服务提供者以服务的参数向 SRS 发去加入申请报文(Join_Probe)，申请加入 WSST；

step2 SRS 接受到申请报文(Join_Probe)后，它根据服务的统一编址方法确定此服务所属的权威，负责解析 SRS 节点，此 SRS 在表中查找此服务所属的的 WSST 的地址，返回给申请服务加入者(joiner)一棵 WSST 中的服务节点地址，或者是此类服务还未有服务注册；

step3 如果 joiner 得到是 WSST 地址，则继续 step4，否则它自己成为此类 WSST 的第一个节点，也是令牌节点，则加入过程结束；

Step4 joiner 向 WSST 连接加入，它首先发出一个报文(还是 Join_Probe)，当 WSST 中某节点 v 接受到一个探测加入报文 Join_Probe；则返回给发起者一个肯定应答报文(Ack_For_Probe)，表示通过此节点加入到 WSST 中；

Step5 WSST 中的节点接受了新服务的加入，接受节点 v 向此类树中拥有令牌的节点发起通告报文，表示树中增加了一个服务；

Step6 令牌的节点在下一次广播报文中通告树中所有节点，整个加入过程结束。

3.3 服务生成图的生成算法

服务生成图的形成需要在工作流语义环境下进行，也就是需要工作流模式的支持。主要思想是，在工作流模式的支持下，工作流起始服务对应的生成树引起生成图的生成，它依据语法关系得知其“下游”的服务类，向 SRS 查询得到其“下游”的生成树地址，进而将“下游”服务联系起来，经过反复的这样查询与连接过程，就形成了一个服务生成图。

算法 2 服务生成图的构造算法

step1 工作流起始服务通过工作流模式得知“下游”服务的参数，以此向 SRS 发去查询此服务的查询申请报文(Query_Probe)；

step2 SRS 接受到查询申请报文(Query_Probe)后，它根据服务的统一编址方法确定此服务所属的权威负责解析 SRS 节点，此 SRS 在表中查找此服务所属的的 WSST 的地址，返回给查询服务者 WSST 中的服务节点地址，或者是空，表示还未有此服务；

step3 如果发起者得到是 WSST 地址，则继续 step4，否则，过程结束；

Step4 发起者向 WSST 连接加入，它首先发出一个报文(Join_Probe)，当 WSST 中某节点 v 接收到一个探测加入报文 Join_Probe，向令牌节点转发，由令牌节点返回给发起者一个肯定应答报文(Ack_For_Probe)，表示此“上下游”关系已经建立；

Step5 新加入 WSST 中的令牌节点接受上游传过来的工作流模式，决定是否继续重复以上步骤，还是工作流关系建立完成。

3.4 Web 服务生成树的信息维护算法

WSST 树内的维护主要是了解树内所有服务的信息，为系统的可用性算法提供基础信息，同时系统还可以根据得到的这些信息来进行负载均衡地提供服务。

维护过程由树内拥有令牌的节点发起，主要过程为信息收集过程和信息分发过程。对于信息收集过程来说，令牌的节点向所有与它相连接的节点发出服务信息，维护广播呼叫报文，各相连节点将自己的服务信息填充到报文后，向所有未接受信息的路径方向广播，而一个节点已经收到过广播后，它删除后面所有发给它的报文，当一个节点没有路径可广播时，它向令牌节点回送报文。对于信息分发过程，当信息收集过程完成后，令牌节点已经收集到了生成树中各节点的信息，它加以综合后，又采用以上的过程向所有节点分发。服务信息交换的伪算法如下。

算法 3 服务信息交互算法

Step1 令牌的节点向所有相邻节点发起服务信息收集报文。

collection_pkg

for (WSST 中每一个接收到报文 collection_pkg 的服务节点)

{ 将自己服务信息加入到报文 collection_pkg 中;

将报文向所有未接受报文的方向转达发;

if (某服务节点的所有邻居都已经转发过了 collection_pkg 报文)

将 collection_pkg 报文向令牌的节点返回; }

Step2 令牌的节点计算所有返回报文的信息。

令牌的节点向所有相邻节点发起服务信息汇总报文 distribution_pkg;

for (每一个接受到报文的节点)

{ 接受报文信息;

将报文向所有未接受 distribution_pkg 报文的方向转达发;}

4 可用性计算

4.1 服务生成树的可用度计算

根据前面的定义，本节使用概率模型来度量服务工作流的可用性。

对于某一类应用对应的 Web 服务集合 $WSST_{ws}=\{WS_1,WS_2,\ldots,WS_m\}$，在假设同时出现 2 个或更多个服务失效的概率是零的情况下，由于系统运行状态具有无后效性，并且失效与重新生成服务状态的转换只限于相邻状态，因此其失效服务个数可以用生灭过程来表示：一个服务出现故障(Birth)将导致状态数加 1，而一个服务生新生成(Death)将导致状态数减 1。

当系统的状态处于n时，表示系统有n个服务处于失效状态；如果一个服务出现失效的到达率为λ_n，某个服务节点重新正常的到达概率为μ_n，那么生灭过程处于状态n的稳态概率p_n如下：

$$P_n=\frac{\lambda_0\lambda_1\ldots\lambda_{n-1}}{\mu_1\mu_2\ldots\mu_n}P_0, n=1,2,\ldots,k,\ldots,\infty \quad (1)$$

其中，P_0为系统处于状态 0 的稳态概率。

针对服务集合$WSST_{ws}$，可以做如下模型假定：(1)服务失效与重新服务的到达时刻都是指数分布分别为λ_n和μ_n；(2)对每个服务在时刻(t,t+Δt)发生失效的条件概率是λΔt；(3)对每个在时刻(t,t+Δt)内重新服务的条件概率是μΔt；(4)同时出现两个或更多个服务节点失效的概率是零；(5)每个服务的失效与重新服务的事件与其它所有事件无关。

这样就可以建立服务集合$WSST_{ws}=\{WS_1,WS_2,..,WS_n\}$的可用度模型。服务集合由n个服务组成，其状态n的稳态概率p_n就是服务集合中所有节点都失效的概率，即整个系统不可用的概率，而A=(1-p_i)即为系统有n-i个服务可用情况下的可用度，实际上表示当服务集合中服务个数为i时系统所具有的可用度为A。再根据下式：

$$\begin{cases} P_k=\dfrac{\lambda_0\lambda_1,...,\lambda_{k-1}}{\mu_1\mu,...,\mu_k}P_0=\dfrac{n(n-1),...,(n-k+1)}{2*3*,...,*k}\left(\dfrac{\lambda}{\mu}\right)^k P_0=C_n^k\left(\dfrac{\lambda}{\mu}\right)^k P_0 \\ \sum\limits_{k=0}^{n} P_k=1 \end{cases} \tag{2}$$

可以得到生成树处于每个状态的稳态概率[8]。求解得

$$P_0=\left(\frac{\mu}{\mu+\lambda}\right)^n \tag{3}$$

这样，生成树处于状态n的稳态概率p_n为

$$P_n=C_n^n\left(\frac{\lambda}{\mu}\right)^n P_0=\left(\frac{\lambda}{\mu}\right)^n\left(\frac{\mu}{\lambda+\mu}\right)^n=\left(\frac{\lambda}{\lambda+\mu}\right)^n \tag{4}$$

此时生成树的可用度为

$$A_n=1-P_n=1-\left(\frac{\lambda}{\mu+\lambda}\right)^n \tag{5}$$

4.2 服务工作流的可用度计算

对于工作流来说，可用度的计算公式比较复杂，如图 1 所示的工作流模式对应的每一类服务类型都对应于网络上的一棵生成树，由于生成树中的任何一个服务都可以作为工作流中的一个执行点，每个服务提供节点的可用度不同，工作流执行路径中常存在复杂的并行与先决条件，并没有统一的原则能够化解这种复杂性，也没有统一的方式来指导这种可用度的计算。

服务在分布式环境中往往是动态变化的，信息的准确与及时性是可用度计算的基础。基于以上的结构模型，本文提出了一种可用度的计算方法，主要思想采用反演的方法，从终点开始向后反演推算，具体实现如下：

算法 4 服务工作流可用度信息获取算法

step1 从工作流模式中对应的终点服务生成树的令牌节点开始进行如下工作；

step2 将自己生成树的可用度信息(假设为 p)反向在所有“上游”服务生成树中传播；

step3 “上游”服务生成树收到所有“下游”服务生成树的可用信息后，依据服务工作流的控制关系 φ 的工作程序如下：

switch(parse(φ))

{//根据不同的控制关系计算不同的可用度，假设本生成树的可用度为 q

case sequence: A=q*p;

//顺利关系，即整个工作流的可用度为 q*p

case while: A= p*; //每循环一次可用度相乘一次

case switch: A=p* φ ;// φ 为某服务为真的概率

case flow: A=(p_i*); //每个服务的p都相相乘

case pick: A= p* ϕ ; // ϕ 为某服务为真的概率

case ▷ : A=1-(1-p_i)*; }

step4 “上游”服务生成树继续向上传播，直到起始服务，则得到整个工作流的可用度 A 及其它可用参数。

5 服务工作流可用性算法

依据前面的知识，现在可以给出完整的服务工作流可用性算法。算法的思想是先根据给出工作流的起始服务，利用工作流的起始服务得到此工作流的服务生成图，然后将生成图中的每一棵生成树施行算法 3，再对生成图施行算法 4，利用算法 4 得到的可用度 A 以及其它可用参数判断系统是否满足需求，如果不需足，则首先增加每一个达不到可用服务个数的生成树中的服务个数(根据式(5)得到应该具有的服务个数，这时再决定是否需要复制或者备份服务，以及复制或者备份服务的个数)，使其达到应用的需求。

这时工作流的整体可用性并不一定满足，继续以上步骤增加可用度最低的生成树的服务个数，直到整个工作流满足应用的需求。

算法 5 服务可用性算法

Step1 以工作流起始服务的参数向 SRS 查询，得到 WSST 令牌的节点(token-node)的 IP 地址；

step2 对生成图中的每一棵生成树执行算法 3；

step3 计算每一棵生成树的可用度参数判断是否达到，如果没有达到，则利用式(5)计算应该增加的服务个数，并增加冗余的服务个数；

Step4 对生成图执行算法 4，如果整个工作流达到应用的需求，则完成,否则转 Step5；

Step5 增加工作流路径中可用度最低的生成树的服务冗余数，再次执行算法 4，直到满足应用的需求。

6 实例分析

本节介绍该算法在网络模拟应用中的工作过程，考虑在全球性广域网中的某一类应用，对于这样一个动态变化的网络，采用 High-Workflow 结构模型后，有如下结论：

现分别假设服务失效后重新生成的平均时间为 50h(2 天多)，每小时平均失效的概率为 0.1(1 天要失效 2 次多)；服务失效后重新生成的平均时间为 30h(1 天多)，每小时平均失效的概率为 0.1(1 天要失效 2 次多) 2 种情况。

对于这样一个动态的网络，如果网络中有 100 个属于同一类应用的服务(即一个生成树)，则采用本文的模型后,计算数据如图 3 所示。图 3 表示尽管在这样一种极其动态的网络中，采用本文的结构模型，此类服务的整体可用性依然很高，具有结构的优越性。

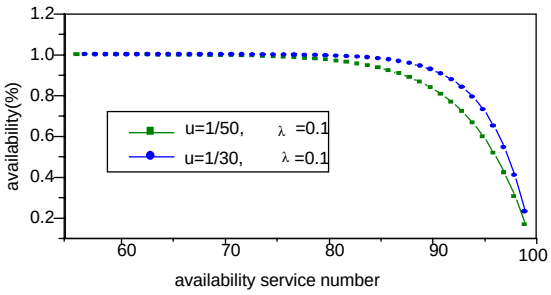

图 3 可用服务个数与此服务类的可用度

如果在应用要求服务的个数不少于 90 个，服务的可用性不低于 95%条件下，按照算法 3 可以计算得到需要复制或者备份的服务数，如表 1 所示。

表 1 需要增加服务的个数

	需要增加的服务数
u=1/50，λ=0.1 的情况	17
u=1/30 λ=0.1 的情况	12

（下转第 42 页）