融合容错需求和资源约束的云容错服务适配方法

杨 娜 刘 靖
(内蒙古大学计算机学院 呼和浩特 010021)

摘 要 云计算环境下,失效成为一种常态行为,可靠性保障能力不足不仅成为云计算应用推广的主要障碍,而且还促使云计算环境下的容错服务研究成为一个亟待解决的问题。针对目前云计算容错服务研究中存在的用户容错需求定义无法直接反映用户关心的可靠性问题,以及云容错服务供应商资源得不到灵活利用等问题,提出了一种融合容错需求和资源约束的云容错服务适配方法。从用户的角度,以组件为单位,基于可靠性对用户的容错需求进行定义。从云容错服务供应商的角度,分别在其资源充足和资源不足的情况下研究最佳的容错服务适配方法,并使用最优化理论求解该适配方法下的容错服务。实验结果表明,所提出的容错服务适配方法生成的容错服务能更好地满足用户和云容错服务供应商的需求。

关键词 云计算,容错即服务,副本容错,检查点容错,最优化
中图法分类号 TP302.8 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.07.011

Cloud Fault Tolerance Services Adaption Method Based on Requirement and Resource Constriction

YANG Na LIU Jing
(College of Computer Science, Inner Mongolia University, Hohhot 010021, China)

Abstract In the environment of cloud computing, faults have become a normal behavior and the shortage of reliability safeguard not only has become the main obstacle of application promotions in cloud computing, but also has made fault tolerance services in cloud computing become a problem that need to be solved urgently. Aiming at solving the shortage study of fault tolerance services in cloud computing that the definition of user fault tolerance requirements can't reflect reliability which was concerned by the users directly and the inflexible usage of the resources of cloud fault tolerance service providers, this paper proposed an adaption method of cloud fault tolerance services which was based on user requirement and resource constriction. This paper first defined the fault tolerance requirements of users from the prospective of users, which was conducted by taking a component as a unit and reliability as a basis. Then the optimal adaption method of fault tolerance services was studied from the perspective of cloud fault tolerance services providers under the condition of that the resources of fault tolerance service providers is insufficient or sufficient. This paper solved the fault tolerance services generated by the optimal adaption method using optimization theory. The results of the experiments showed that the fault tolerance services which were generated by our adaption method can better satisfy the requirements of users and cloud fault tolerance service providers.

Keywords Cloud computing, Fault tolerance as a service, Replication fault tolerance, Checkpoint fault tolerance, Optimization

1 引言

云计算底层基础设施的高度复杂性使得数据中心遭受大量的故障,同时也降低了云应用系统的可靠性。容错能保证系统即使在遭受故障的情况下也能正确持续地运行,对云用户和云服务供应商而言十分重要[1]。

云计算环境下,单独由云基础架构服务提供商为上层运行的各个云应用系统提供可用的容错机制,很难顾全不同云应用系统的容错需求;同时,单独由云应用系统研发及运营商为其实现可用的容错机制,也很难充分利用底层的云基础机构资源和效能。目前,由第三方容错服务提供商以容错即服务的模式为云应用系统的多个用户提供持续可用的容错服务,已成为首选的容错服务实用模式。其优势主要在于能够充分利用底层云资源来实现不同级别的容错服务,并根据上层不同云应用系统用户的容错需求,为其适配并集成不同的容错服务,为该用户提供了定制的容错服务,且同时易于保障容错服务提供商的利益最大化。

对容错即服务的研究大多都是假设用户在定义容错需求时指定容错方法,但此做法存在不足。一方面,容错解决的主要问题是提高应用的可靠性,可靠性是用户最关心的问题;让

到稿日期:2016-08-20 返修日期:2016-11-07 本文受国家自然科学基金项目(61262017,61662051)资助。
杨 娜(1991-),女,硕士生,主要研究方向为云软件容错,E-mail:yangna_sd@126.com;刘 靖(1981-),男,博士,副教授,CCF高级会员,主要研究方向为云计算、容错技术、软件可靠性工程,E-mail:liujing@imu.edu.cn(通信作者)。

用户指定容错方法会使用户的容错需求无法直接反映用户关心的可靠性问题。另一方面，如果用户指定容错方法，则满足用户容错需求的资源是固定的；云容错服务供应商现有资源的固定性以及满足用户容错需求的资源的固定性，会使云容错服务供应商只能为固定数量的用户提供容错服务，这不仅会导致云容错服务供应商因资源得不到灵活利用而减少收益，而且还会因云容错服务供应商的资源不足而导致云容错服务供应商拒绝为某些用户提供服务；用户一旦被拒绝提供服务，则可能会在后续时间里转向别的供应商请求服务，从而降低该容错服务供应商的后续的客户量，进而对云容错服务供应商后续收益产生一定的影响。

针对以上容错即服务研究中存在的不足，本文从用户的角度分析用户的容错需求，以组件为单位对其进行基于可靠性和可靠性误差的定义。从云容错服务供应商的角度研究了两种情况下最佳的容错服务适配方法，并使用最优化理论求解了该适配方法下的容错服务。当云容错服务供应商的资源足够时，综合考虑用户和云容错服务供应商的利益，选择使云容错服务供应商开销最低的容错服务适配方法作为最佳的容错服务适配方法，此时云容错服务供应商的成本和用户支付的费用较低。当云容错服务供应商的资源不足时，允许云容错服务供应商将某些组件的可靠性降低至用户允许的误差范围内，或外包资源，而后综合考虑用户和云容错服务供应商的利益，选择使云容错服务供应商利益最大的容错服务适配方法作为最佳的适配方法。此时，不仅云容错服务供应商的成本低，而且用户只需支付较低的费用即可得到较好的服务质量。该方法能够使云容错服务供应商在满足用户容错需求的同时，通过灵活利用现有资源保证云容错服务供应商的客户量并提高收益。

本文第 2 节介绍了目前分布式计算环境中关于容错和容错服务的研究；第 3 节从应用的角度研究了应用的容错需求定义；第 4 节计算了使用不同容错方法满足用户容错需求时的资源需求和开销；第 5 节分别在云容错服务供应商资源充足和资源不足的情况下研究了最佳的容错服务适配方法，并求解了该容错适配方法下的容错服务；第 6 节使用 CloudSim 和 Lingo 进行实验，验证了本文所提容错服务适配方法的优势；最后总结本文并展望未来。

2 相关工作

文献[2]提出了云计算环境下动态可适配的副本容错方法和动态可适配的检查点容错方法，并对这两种方法进行了分析、建模和评估；文献[3]针对当前计算机系统计算和存储资源丰富但并行文件系统写带宽提高相对滞后的特点，提出了基于内存缓存的异步检查点容错技术；文献[4]针对云计算系统的软故障恢复，提出了能源感知的容错调度框架；文献[5]为分布式组件系统提出了一种容错框架；文献[6]对节能存储系统的容错框架进行了研究；文献[7]提出将容错作为服务层的概念架构，其属于框架级设计，并没有提出具体的容错即服务方案和容错方法实施架构；文献[8]从容错服务供应商的角度，研究了选择为哪些租户提供容错服务能使云容错服务供应商的收益最大，用户必须指定组件将要使用的容错方法，容错方法所需资源的固定性会使云容错服务供应商的资源得不到灵活利用。

在容错服务的研究中，从用户的角度对用户的容错需求进行详细的分析和定义，并从云容错服务供应商的角度基于用户容错需求和云容错服务供应商的现有资源为用户提供可适配的容错服务，以便使得云容错服务供应商的收益最大，变得十分重要。

3 用户容错需求定义

本文讨论的容错服务主要是为云环境中 SaaS 层的云应用软件提供可靠性保障。本文中的用户是特指云应用软件本身即云应用软件作为第三方容错服务的用户。下面从用户的角度分析用户的容错需求。

从用户的角度来看，应用中不同组件的重要程度不同(本文中的组件是指具有一定功能的独立程序体)。重要性越高的组件发生故障后的损失越大，用户更倾向于为重要程度较高的组件定义较高的容错需求，所以用户希望容错服务供应商以组件为单位提供容错服务。

组件在不同时间段的重要程度可能不同，如某个组件在某个时间段被调用的次数较多，此时该组件的重要性高于其他时间段的重要性，所以用户希望容错服务供应商以时间段为单位提供服务。

可靠性是用户最关心的问题。由于一些非关键性组件对可靠性无严格要求，定义可靠性误差使用户的容错费用存在降低的可能，所以用户希望容错服务供应商基于可靠性和可靠性误差提供服务。

通过以上分析，用五元组表示组件的容错需求：

$\langle AID_CID, periodT, Critic_{AID_CID}(periodT), URelia_{AID_CID}(periodT), DRelia_{AID_CID}(periodT)\rangle$

对该五元组进行如下说明。

1)AID_CID：表示组件的标识。其中，AID 表示应用的标识，CID 表示应用中组件的标识，且 $AID\in N^+$，$CID\in N^+$。

2)$periodT$：表示组件需要云容错服务供应商提供容错服务的时间段。

3)$Critic_{AID_CID}(periodT)$：表示组件 AID_CID 在时间段 $periodT$ 内的关键性。其中 $Critic_{AID_CID}(periodT)\in\{0,1\}$，取值为 0 时表示组件 AID_CID 在时间段 $periodT$ 内是非关键性组件；取值为 1 时表示组件 AID_CID 在时间段 $periodT$ 内是关键性组件。

4)$URelia_{AID_CID}(periodT)$：表示组件 AID_CID 在时间段 $periodT$ 内的可靠性需求。其中，$URelia_{AID_CID}(periodT)<1$。

5)$DRelia_{AID_CID}(periodT)$：表示组件 AID_CID 在时间段 $periodT$ 内允许的可靠性误差。其中，$DRelia_{AID_CID}(periodT)\geqslant 0$。

4 容错方法的资源需求和开销计算

4.1 容错方法的开销说明

提供容错服务需要基于特定的容错方法，并考虑不同容错方法的开销，以便选择更为适合的容错方法来提供容错服务。本文讨论当前被普遍使用的副本容错方法、检查点容错方法和 NVP 容错方法。因为大部分的容错方法主要是通过增加额外的硬件、软件来提高系统的可靠性，所以在考虑容错

方法带来的开销时，主要考虑不同的容错方法在虚拟机、存储资源和软件运行方面的开销。

特定时间段内使用虚拟机产生的费用作为该时间段的虚拟机开销，特定时间段内使用存储资源的费用作为该时间段的存储资源开销。此外，检查点容错方法除了虚拟机和存储资源开销外，还有生成及应用检查点引起的软件运行开销。根据文献[8]，把使用检查点引起的成本作为检查点容错方法在软件运行方面的开销。

计算总开销时，因为不同时间段云容错服务供应商对各种开销的关注程度可能不同，如某个时间段云容错服务供应商可能更关心虚拟机开销，所以在计算总开销时为各种开销增加了权重因子，这样云容错服务供应商即可根据不同的场景需求对不同的开销设置不同的权重。在此设置 α 为存储开销权重因子，β 为虚拟机开销权重因子，γ 为其他开销的权重因子。

4.2－4.4 节将以组件在时间段[0，T]（单位为小时）内的可靠性需求 $URelia_i(T)$ 为例，计算在该可靠性需求下使用不同容错方法时的资源需求和开销。假设运行程序（在副本和 NVP 容错方法中指组件本身，在检查点容错方法中指检查点程序）所需的存储单元大小为 n_m，组件自身的初始可靠性为 R，数据中心的大小为 s，数据中心的调节系数为 u。

4.2　副本容错方法的资源需求和开销

副本容错方法是对系统关键部分的组件进行备份，以便在系统关键部分发生故障后保证至少存在一个副本可用。这里考虑主动副本容错技术即组件的不同副本部署在不同虚拟机上，并且每个副本同时处理相同的请求。用户发出请求后，通过一定的算法选择某个虚拟机上的副本的执行结果返回给用户。当选中的副本运行失败时，通过算法选择另一个副本提供相同的服务。可利用式(1)计算出，在时间段[0，T]内，使用副本容错方法满足组件 i 的可靠性需求 $URelia_i(T)$ 时至少需要的副本个数，即 $numRep$ 的值。

$$1-(1-R)^{numRep}\geqslant URelia_i(T) \tag{1}$$

计算副本容错方法的资源需求时，根据副本容错方法的方法机制和式(1)可知，使用副本容错方法且满足用户可靠性需求时，由于容错方法而使用的资源为 $numRep-1$ 台虚拟机（计算资源）和 $numRep-1$ 个存储单元（存储资源）。

计算存储开销时，根据副本容错方法使用的存储单元的数量，并参考文献[9]提出的云计算下的数据管理成本模型，采用式(2)表示在时间段[0，T]内副本容错方法产生的存储开销。

$$Cost_{storRep}=\frac{u}{2\sqrt{s}}n_m(numRep-1)T \tag{2}$$

计算虚拟机开销时，根据副本容错方法使用的虚拟机数量，使用式(3)表示副本容错方法产生的虚拟机开销。这里用 $priceVmHour$ 表示云容错服务供应商每台虚拟机每小时的价格。

$$Cost_{vmRep}=(numRep-1)priceVmHour\cdot T \tag{3}$$

根据以上分析，在时间段[0，T]内使用副本容错方法产生的总开销如式(4)所示。

$$Cost_{sumReplication}=\alpha Cost_{storRep}+\beta Cost_{vmRep} \tag{4}$$

4.3　NVP 容错方法的资源需求和开销

N 版本编程技术(NVP)系统，是通过对冗余模块的输出执行表决以屏蔽子系统产生的错误。在 NVP 容错技术中，当用户请求到来时，n 个版本的子系统同时执行（子系统运行在不同的虚拟机上），子系统的执行结果传输到表决器，表决器根据表决算法确定处理是否正确。文献[10]已对表决器的表决算法做了详细的分析。本文使用被普遍使用的大数表决算法，即在 n 个冗余输入中仅当至少一半或者一半以上的结果一致时才认为该结果是正确的。由式(5)可计算出，在时间段[0，T]内，满足组件 i 的可靠性需求 $URelia_i(T)$ 时，至少需要的子系统个数，即 $numNVP$ 的值。

$$\sum_{j=\frac{numNVP}{2}+1}^{numNVP}C_{numNVP}^{j}R^{j}(1-R)^{numNVP-j}\geqslant URelia_i(T) \tag{5}$$

计算 NVP 容错方法的资源需求时，根据 NVP 容错方法的方法机制和式(5)可知，使用 NVP 容错方法且满足用户可靠性需求时，由于容错方法而使用的资源为 $numNVP-1$ 台虚拟机和 $numNVP-1$ 个存储单元。

计算存储开销时，根据 NVP 容错方法使用的存储单元的数量，并参考文献[9]中的云计算下的数据管理成本模型，采用式(6)表示在时间段[0，T]内 NVP 容错方法产生的存储开销。

$$Cost_{storNVP}=\frac{u}{2\sqrt{s}}n_m(numNVP-1)T \tag{6}$$

计算虚拟机开销时，根据 NVP 容错方法使用的虚拟机数量，采用式(7)表示 NVP 容错方法产生的虚拟机开销。

$$Cost_{vmNVP}=(numNVP-1)priceVmHour\cdot T \tag{7}$$

根据以上分析，在时间段[0，T]内，由于使用 NVP 容错方法而产生的总开销如式(8)所示。

$$Cost_{sumNVP}=\alpha Cost_{storNVP}+\beta Cost_{vmNVP} \tag{8}$$

4.4　检查点容错方法的资源需求和开销

检查点容错方法是通过计算机系统周期性地保存一个进程在非失效时的运行状态，使该进程在失效后，能从已保存的状态处继续执行，而不必重新开始，这样可以减少因失效而损失的计算量，从而达到容错的目的[11]。本文考虑最基本的完全检查点容错方法，它主要是周期性地保存整个系统的状态到存储平台。当系统发生故障时，系统恢复到最新的检查点处并继续运行。

计算检查点容错方法的资源需求时，根据检查点容错方法的机制，使用检查点容错方法且满足用户可靠性需求时，由于检查点容错方法而使用的资源为 1 台虚拟机和 1 个存储单元。

计算存储开销时，根据检查点容错方法使用的存储单元数量，并参考文献[9]中云计算下的数据管理成本模型，采用式(9)表示在时间段[0，T]内检查点容错方法产生的存储开销。

$$Cost_{storCp}=\frac{u}{2\sqrt{s}}n_m\cdot 1\cdot T \tag{9}$$

计算虚拟机开销时，根据检查点容错方法使用的虚拟机资源数量，采用式(10)表示检查点容错方法产生的虚拟机开销。

$$Cost_{vmCP}=1\cdot priceVmHour\cdot T \tag{10}$$

计算该时间段内由于软件运行产生的开销时，先根据4.1节计算在该时间段内由于使用检查点而产生的开销。首先根据文献[8]计算在时间段[0，T]内检查点的个数，然后计算该

时间段内检查点的个数与每使用一次检查点产生的开销的乘积，并将其作为该时间段内检查点容错方法在使用检查点方面的开销。下面计算检查点容错方法在这方面的开销。

首先假设设置检查点需要的时间为 T_{chk}；然后根据文献[2]，用式(11)表示当组件 i 在时间段$[0,T]$内的可靠性需求为 $URelia_i(T)$时，检查点的时间间隔为 ΔT；最后采用式(12)表示在时间段$[0,T]$内使用检查点容错方法时，由于使用检查点而引起的开销。采用 costPerInnovation 表示使用一次检查点引起的开销。

$$\Delta T=\left(\frac{T_{chk}}{\ln\frac{1}{URelia_i(T)}}\right)^{0.5} \tag{11}$$

$$Cost_{otherCP}=\frac{T}{\Delta T}\cdot costPerInnovation \tag{12}$$

通过以上分析，在时间段$[0,T]$内使用检查点容错方法产生的总开销如式(13)所示。

$$Cost_{sumCP}=\alpha Cost_{storCP}+\beta Cost_{vmCP}+\gamma Cost_{otherCP} \tag{13}$$

本节对使用不同容错方法满足组件可靠性需求时需要的资源和引起的开销做了计算；下节将首先给出容错服务的定义，然后生成最佳的容错服务适配方法，最后以资源需求和开销为基础求解该适配方法下的容错服务。

5 最佳的容错服务适配方法

5.1 容错服务的定义

组件 i 在时间段$[0,T]$内提供的容错服务用以下七元组进行表示。

$\langle Rep_i(T), NVP_i(T), CP_i(T), R_i(T), LRelia_i(T), FRelia_i(T), Cost_{fee(i)}^{Relia}(T)\rangle$

对该七元组分别作如下说明。

(1)$Rep_i(T)$表示在时间段$[0,T]$内，是否为组件 i 应用副本容错方法。

$$Rep_i(T)=\begin{cases}1, & 是\\ 0, & 否\end{cases}$$

(2)$NVP_i(T)$表示在时间段$[0,T]$内，是否为组件 i 应用 NVP 容错方法。

$$NVP_i(T)=\begin{cases}1, & 是\\ 0, & 否\end{cases}$$

(3)$CP_i(T)$表示在时间段$[0,T]$内，是否为组件 i 应用检查点容错方法。

$$CP_i(T)=\begin{cases}1, & 是\\ 0, & 否\end{cases}$$

(4)$R_i(T)$表示在时间段$[0,T]$内，组件 i 是否是关键性组件。

$$R_i(T)=\begin{cases}1, & 是\\ 0, & 否\end{cases}$$

(5)$LRelia_i(T)$表示在时间段$[0,T]$内，组件 i 的可靠性需求是否被降低。

$$LRelia_i(T)=\begin{cases}1, & 是\\ 0, & 否\end{cases}$$

(6)$FRelia_i(T)$表示在时间段$[0,T]$内，为组件 i 应用容错服务时最终达到的可靠性的值。

$$FRelia_i(T)=\begin{cases}URelia_i(T), & LRelia_i(T)=0\\ URelia_i(T)-DRelia_i(T), & LRelia_i(T)=1\end{cases}$$

(7)$Cost_{fee(i)}^{Relia}(T)$表示在时间段$[0,T]$内，组件 i 被服务的可靠性为 $Relia$ 时支付给云容错服务供应商的费用。组件可靠性没有被降低时，使用云容错服务供应商实际分配给组件 i 的容错服务所带来的总开销作为组件 i 应该支付给云容错服务供应商的费用。组件可靠性被降低时，使用云容错服务供应商实际分配给组件的容错服务所带来的总开销的一部分作为组件 i 应该支付给云容错服务供应商的费用(后面用 w 表示组件可靠性被降低后实际支付的费用占实际总开销的比重)。

5.2 节和 5.3 节将分别在资源足够和资源不足的情况下生成最佳的容错服务适配方法，并求解最佳的容错服务适配方法生成的容错服务。

5.2 云容错服务供应商资源足够时生成最佳的容错服务适配方法

5.2.1 资源足够时容错服务适配方法的生成

由 5.1 节中定义的组件支付给云容错服务供应商的费用的支付规则可知，应该为组件提供开销低的容错服务。因为开销低的容错服务会使组件支付的费用低，使得更多的组件购买该容错服务供应商的服务而不去购买其他容错服务供应商的服务，这间接地增大了云容错服务供应商的客户量，从而增加了云容错服务供应商的收益。所以在这种情况下，使云容错服务供应商总开销最低的容错服务适配方法就是最佳的容错服务适配方法。

5.2.2 资源足够时容错服务的求解

使用最优化理论求解最佳的容错服务适配方法所生成的容错服务。由于在时间段$[0,T]$内为每个组件只分配一种容错方法，因此容错服务应满足约束条件(14)。

$$Rep_i(T)+NVP_i(T)+CP_i(T)=1 \tag{14}$$

其中，$1\leqslant i\leqslant p$(p 是指组件个数)。

因为这种情况下云容错服务供应商的虚拟机资源和存储资源是足够的，所以组件的可靠性无需被降低，即 $LRelia_i(T)$的值恒为 0，因此容错服务应满足约束条件(15)。

$$FRelia_i(T)=URelia_i(T) \tag{15}$$

其中，$1\leqslant i\leqslant p$(p 是指组件个数)。

根据该节分析，在时间段$[0,T]$内定义目标函数如式(16)所示。

$$P_{enough}=\min\left(\sum_{i=1}^{p}(Rep_i(T)Cost_{sumRep}^{FRelia_i(T)}(T)+NVP_i(T)Cost_{sumNVP}^{FRelia_i(T)}(T)+CP_i(T)Cost_{sumCP}^{FRelia_i(T)}(T)\right) \tag{16}$$

其中，$Cost_{sumRep}^{FRelia_i(T)}(T)$ 指的是组件在时间段$[0,T]$内应用副本容错方法且达到可靠性 $FRelia_i(T)$时的开销，其计算过程与 4.2 节中计算开销的方式相同；同理可得 $Cost_{sumNVP}^{FRelia_i(T)}(T)$ 和 $Cost_{sumCP}^{FRelia_i(T)}(T)$ 的计算方式。

使用目标函数(16)以及约束条件(14)和(15)建立数学模型，求解数学模型的可行解。通过分析可行解可得为各组件提供的容错服务。

5.3 云容错服务供应商资源不足时生成最佳的容错服务适配方法

根据第 3 节中对用户容错需求定义的分析可知，云容错

服务供应商在提供容错服务时，可将非关键性组件的可靠性降低至用户允许的误差范围内，并且可以外包资源。

容错方法的不固定性、可靠性误差的设置和资源的外包，使得资源利用具有灵活性，可提高云容错服务供应商的收益。例如，当前云容错服务供应商在现有资源状况下不能为某个组件提供容错服务，但是，外包一台虚拟机后，因为用户容错方法的非固定性，云容错服务供应商就有可能通过充分利用现有资源来重新生成新的容错服务以满足该组件的容错需求。此时，云容错服务供应商因为满足该组件的容错需求而增加的收益就会大于外包资源的费用（增加的收益至少为一台虚拟机和一个存储单元的费用）。同理，云容错服务供应商也可能在外包资源的同时降低某些组件的可靠性，释放一些资源，然后再通过充分利用现有资源来重新生成新的容错服务以满足更多用户的容错需求，进而获得更好的收益。基于以上对资源利用灵活性的分析，假设云容错服务供应商在为所有组件提供容错服务之前，外包资源引起的成本低于云容错服务供应商因为外包增加的资源使得能提供更好的容错服务而多得的收益。

5.3.1　资源不足时容错服务适配方法的生成

因为用户在为所有组件提供容错服务之前，外包资源引起的成本低于容错服务供应商因为外包增加的资源使得能提供更好的容错服务而多得的收益，所以对云容错服务供应商而言，为所有组件提供容错服务时的收益是最大的。

根据 5.1 节中定义的组件给云容错服务供应商支付费用的支付规则可知，应该为组件提供开销低的容错服务。因为开销低的容错服务会使组件支付的费用低，此时更多的组件就会购买该容错服务供应商的服务而不去购买其他容错服务供应商的服务，这会间接地提高云容错服务供应商的客户量，从而提高云容错服务供应商的收益。所以，从云容错服务供应商和用户的角度来看，云容错服务供应商应该尽量为组件提供成本低的容错服务。

因为允许云容错服务供应商降低某些组件的可靠性，所以在此情况下，云容错服务供应商会以开销低（用户支付费用低）为目标把所有组件的可靠性降低（可靠性降低后，满足可靠性的开销就会降低），这会导致用户的服务质量出现不必要的降低，进而导致云容错服务供应商的收益也会出现不必要的降低。所以，从云容错服务供应商的角度来看，应该尽量减少因为降低组件可靠性而少收取的收益，即尽量保证服务质量。

综合以上分析，对云容错服务供应商而言，既能为所有组件提供容错服务，又能保证组件支付较低的费用，还能保证服务质量的容错服务适配方法能够使云容错服务供应商的收益最大。所以，当云容错服务供应商资源不足时，定义最佳的容错服务适配方法就是在为所有组件提供容错服务的条件下，使得云容错服务供应商的成本与因为降低组件可靠性而少收取的费用的和最低的容错服务适配方法。

5.3.2　资源不足时容错服务的求解

首先给出以下定义，然后求解最佳的容错服务适配方法所生成的容错服务。

(1)定义 $numRep'$ 表示组件可靠性被降低并应用副本容错方法时需要的副本个数。$numNVP'$ 表示组件可靠性被降低并应用 NVP 容错方法时需要的子系统个数，其计算方式与 $numNVP$，$numReplication$ 的计算方式相同，只是可靠性为降低之后的可靠性。

(2)定义云容错服务供应商在时间段$[0,T]$内，外包资源引起的费用为 $outFee(T)$；并用 C_{vm}^{0} 表示外包一台虚拟机的价格，C_{s}^{0} 表示外包一个存储单元的价格；$outV(T)$ 表示在时间段$[0,T]$内外包的虚拟机数量；$outS(T)$ 表示在时间段$[0,T]$内外包的存储单元数量。那么，$outFee(T)$ 如式(17)所示。

$$outFee(T)=C_{vm}^{0}\cdot outV(T)+C_{s}^{0}\cdot outS(T) \tag{17}$$

其中，$1\leqslant i\leqslant p$。

(3)定义 $preVm_i(T)$ 为组件可靠性没有被降低，且组件 i 在时间段$[0,T]$内应用不同容错方法时，容错方法对虚拟机数量的不同需求，如式(18)所示。

$$preVm_i(T)=(numNVP-1)\cdot NVP_i(T)+1\cdot CP_i(T)+(numRep-1)\cdot Rep_i(T) \tag{18}$$

其中，$1\leqslant i\leqslant p$。

(4)定义 $postVm_i(T)$ 为组件可靠性被降低，且组件 i 在时间段$[0,T]$内应用不同容错方法时，容错方法对虚拟机数量的不同需求，如式(19)所示。

$$postVm_i(T)=(numNVP'-1)\cdot NVP_i(T)+1\cdot CP_i(T)+(numRep'-1)\cdot Rep_i(T) \tag{19}$$

其中，$1\leqslant i\leqslant p$。

(5)定义 $preStor_i(T)$ 为组件可靠性没有被降低，且组件 i 在时间段$[0,T]$内应用不同容错方法时，容错方法对存储单元数量的不同需求，如式(20)所示。

$$preStor_i(T)=(numNVP-1)\cdot NVP_i(T)+1\cdot CP_i(T)+(numRep-1)\cdot Rep_i(T) \tag{20}$$

其中，$1\leqslant i\leqslant p$。

(6)定义 $postStor_i(T)$ 为组件可靠性被降低，且组件 i 在时间段$[0,T]$内应用不同容错方法时，容错方法对存储单元数量的不同需求，如式(21)所示。

$$postStor_i(T)=(numNVP'-1)\cdot NVP_i(T)+1\cdot CP_i(T)+(numRep'-1)\cdot Rep_i(T) \tag{21}$$

其中，$1\leqslant i\leqslant p$。

(7)定义 $loseFee(T)$ 为在时间段$[0,T]$内，因为降低某些组件的可靠性而导致云容错服务供应商少收取的费用，如式(22)所示。

$$loseFee(T)=\sum_{i=1}^{p}Cost_{fee(i)}^{Relia}(T)(1-w)\ LRelia_i(T) \tag{22}$$

(8)定义 $finalCost(T)$ 为在时间段$[0,T]$内，容错服务供应商为组件应用容错方法所产生的总开销，如式(23)所示。

$$finalCost(T)=\sum_{i=1}^{p}Rep_i(T)\,Cost_{sumRep}^{FRelia_i(T)}(T)+NVP_i(T)\,Cost_{sumNVP}^{FRelia_i(T)}(T)+CP_i(T)\,Cost_{sumCP}^{FRelia_i(T)}(T) \tag{23}$$

下文将在上述定义的基础上求解容错服务。

因为在时间段$[0,T]$内必须为每个组件只分配一种容错方法，所以容错服务应满足约束条件(14)。因为只有当组件为非关键性组件时才能降低组件的可靠性，所以在时间段$[0,T]$内容错服务应满足约束条件(24)。

$$LRelia_i(T)+CritiC_i(T)\leqslant 1 \tag{24}$$

因为在时间段$[0,T]$内，所有组件由于容错而使用的虚

拟机数量的和不能超过容错服务供应商现有的虚拟机数量($V(T)$)与外包的虚拟机数量的和,所有组件由于容错而使用的存储单元数量的和不能超过容错服务供应商现有的存储单元数量($S(T)$)与外包的存储单元的数量的和,所以在时间段$[0,T]$内,容错服务必须满足约束条件(25)和条件(26)。根据分析,定义目标函数如式(27)所示。

$$\sum_{i=1}^{p} preVm_i(T)R_i(T)+(postVm_i(T)\cdot LRelia_i(T)+preVm_i(T)\cdot(1-LRelia_i(T)))\cdot(1-R_i(T))\leqslant V(T)+outV(T) \tag{25}$$

$$\sum_{i=1}^{p} preStor_i(T)R_i(T)+(postStor_i(T)\cdot LRelia_i(T)+preStor_i(T)\cdot(1-LRelia_i(T)))\cdot(1-R_i(T))\leqslant S(T)+outS(T) \tag{26}$$

$$P_{notEnough}=\min(loseFee(T)+finalCost(T)) \tag{27}$$

使用目标函数(27)、约束条件(14)以及式(24)—式(26)建立数学模型,求解数学模型可得模型的可行解。通过分析可行解可得为各组件提供的容错服务。

5.1 节—5.3 节定义了组件的容错服务,生成了在云容错服务供应商资源足够和资源不足的情况下最佳的容错服务适配方法,求解了在最佳的容错服务适配方法下为组件提供的容错服务。

6 实验评价

结合本文的实验特点,使用 CloudSim[12] 建立实验所需要的数据中心,并使用 Lingo[13-14] 求解最优的容错服务适配方法所生成的容错服务。首先,对实验所需的数据进行收集和准备;然后,研究在云容错服务供应商资源足够时,固定容错服务(这里的固定指的是容错服务中容错方法的固定)和非固定容错服务(本文提出的最优的容错服务适配方法生成的容错服务)的场景下,云容错服务供应商为所有组件提供容错服务时的总开销和每个组件平均支付的金额;最后,研究了当云容错服务供应商资源不足时,固定容错服务和非固定容错服务的情况下,每个组件平均支付的金额、降低组件可靠性而少收取的费用以及容错服务供应商的代价(容错服务供应商的成本和外包资源的费用之和)。在组件个数为 6,9,19,29 的情况下进行实验。

6.1 实验准备

(1)设置数据中心的数值。按照文献[9]中的实验数据,设置数据中心的大小 s 为 60TB,数据中心的调节系数 u 为 1。

(2)设置组件的容错需求。参照文献[15]中应用组件的可靠性的值,设置本文所需的代表组件容错需求的数据。具体的容错需求中各个元组的数值设置如下:1) $periodT$ 设置为 1,即组件在[0,1]点请求容错服务;2)因为在应用 1 中,任务 4 为保存操作,根据其重要性,将其关键性设置为 1,将应用 1 中其他组件的关键性随机设置为 0 或者 1;3)$URelia$ 设置为 R 与(0,0.01)之间的任一随机数的和,将应用 1 中组件 1(task1)的初始可靠性 R 的值设置为文献[15]实验数据中的任意一个服务组件的平均值,这里设置为 aus 服务组件的平均值即 0.989,设置组件 2—组件 6 的初始组件可靠性为(0.935,1.000)之间的随机数(这里去除了 CN 用户测得的异常点),其中,0.935 为最低的可靠性的值,1.000 为最高的可靠性的值;4)设置应用 1 中各组件的 $DRelia$ 为(0,($URelia-R$))之间的随机数值。根据以上设置,得到应用的各个组件的容错需求,如表 1 所列。

表 1 组件容错需求

periodT	AID_CID	Critic	URelia	DRelia	R
0-1	1-1	1	0.996	0	0.989
0-1	1-2	0	0.966	0.008	0.956
0-1	1-3	0	0.991	0.003	0.983
0-1	1-4	1	0.966	0	0.947
0-1	1-5	0	0.953	0.006	0.944
0-1	1-6	0	0.958	0.001	0.953

(3)设置实验中用到的其他参数的数值。根据亚马逊云平台的资源价格,设置 $priceVmHour$ 为 0.085,即每小时每台虚拟机的费用为 \$0.085;$perCostStorage$ 为 0.15,即 1G 大小的数据每月的存储费用为 \$0.15;C_{vm}^{v} 为 \$0.09(略高于云容错服务供应商现有资源的价格);C_s^0 为 \$0.2(略高于云容错服务供应商现有资源的价格);T_{chk} 为 2s;w 为 0.9;$resultSize$ 为 1;T 为 1;权重因子 α,β,γ 都为 1;在设置 $costPerInvocation$ 的值时,不失一般性,将其设置为在使用一次检查点所需的时间内,一台虚拟机和一个存储单元的费用。假设使用一次检查点所需时间为 2s,那么根据该节中其他参数的值可得:$costPerInvocation=2*(0.085+0.15/(30*24))/(60*60)\approx 0.000048$(按照一个月 30 天计算)。

(4)设置固定容错服务。组件个数为 6 时:第 1,2,5 个组件使用副本容错方法,第 3,6 个组件使用 NVP 容错方法,第 4 个组件使用检查点容错方法。组件个数为 9 时:第 1,2,5 个组件使用副本容错方法,第 3,6,7 个组件使用 NVP 容错方法,第 4,8,9 个组件使用检查点容错方法。组件个数为 19 个时:第 1,2,5,11,12,15 个组件使用副本容错方法,第 3,6,9,10,13,17,19 个组件使用 NVP 容错方法,第 4,7,8,14,16,18 个组件使用检查点容错方法。组件个数为 29 时:第 1,2,5,15,20,26,28 个组件使用副本容错方法,第 3,6,7,10,11,12,14,18,19,22,23,24,29 个组件使用 NVP 容错方法,第 4,8,9,13,16,17,21,25,27 个组件使用检查点容错方法。

6.2 云容错服务供应商资源足够时

实验 1 和实验 2 使用表 1 中的组件的容错需求数据作为组件个数为 6 时实验所需的数据;组件个数为 9,19,29 时,除了使用表 1 中的 6 条数据外,还参考这 6 个组件的容错需求数据随机生成了剩余的数据。

实验 1 探究在不同组件个数下,固定容错服务引起的所有组件的总开销和非固定容错服务引起的所有组件的总开销,结果如图 1 所示。

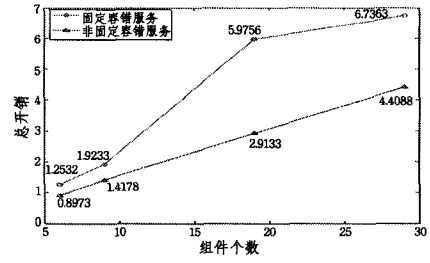

图 1 (资源足够时)所有组件的总开销

从图 1 可以看出,非固定容错服务时云容错服务供应商

的总开销低于固定容错服务时云容错服务供应商的总开销，即非固定容错服务能够更好地满足云容错服务供应商的需求。

实验 2　探究在不同组件个数下，固定容错服务时每个组件平均支付的金额和非固定容错服务时每个组件平均支付的金额，结果如图 2 所示。

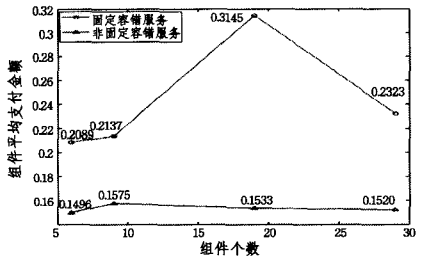

图 2　（资源足够时）组件平均支付的金额

从图 2 可以看出，非固定容错服务时每个组件平均支付的金额低于固定容错服务时每个组件平均支付的金额，即非固定容错服务亦即使用本文提出的最佳的容错服务适配方法生成的容错服务能够更好地满足用户的需求。

通过实验 1 和实验 2 可以得出，当云容错服务供应商资源足够时，本文提出的云容错服务适配方法能够更好地满足云容错服务供应商和用户的需求。

6.3　云容错服务供应商资源不足时

为了使实验效果更加明显，在实验中加大了部分组件的初始可靠性和请求的可靠性之间的差距，将差距改为 0.1 左右。

为了证明在此情况下，所提出的云容错服务适配方法能够在更好地满足用户要求的前提下，提高云容错服务供应商的收益。从组件的角度，探究了固定容错服务和非固定容错服务时，每个组件平均支付的金额和云容错服务供应商为组件提供的服务质量；从云容错服务供应商的角度，探究了固定容错服务和非固定容错服务时云容错服务商为组件提供容错服务所产生的代价。

实验 3　探究在不同组件个数下，固定容错服务时每个组件平均支付的金额和非固定容错服务时每个组件平均支付的金额，结果如图 3 所示。

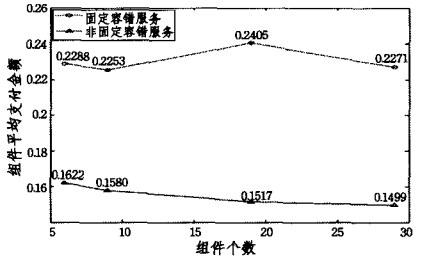

图 3　（资源不足时）组件平均支付的金额

从图 3 可以看出，非固定容错服务时每个组件平均支付的金额低于固定容错服务时每个组件平均支付的金额，即非固定容错服务能够更好地满足用户的需求。

实验 4　探究在不同组件个数下，固定容错服务时云容错服务供应商因为降低组件可靠性而少收取的费用，以及非固定容错服务时因为降低组件可靠性而少收取的费用的大小即 LoseFee 的大小。因为 LoseFee 的大小反映了云容错服务供应商降低的容错服务质量，所以用降低的服务质量表示 LoseFee，结果如图 4 所示。

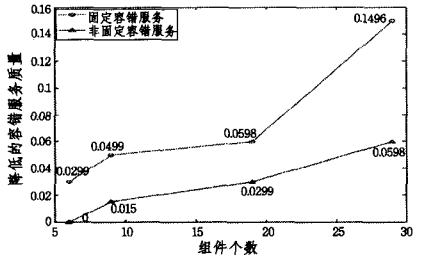

图 4　（资源不足时）容错服务供应商降低的容错服务质量

从图 4 可以看出，非固定容错服务时云容错服务供应商降低的容错服务质量低于固定容错服务时云容错服务供应商降低的服务质量，即非固定容错服务能够更好地保证用户的容错服务质量。

实验 5　探究在不同组件个数下，固定容错服务时云容错服务供应商的代价和非固定容错服务时云容错服务供应商的代价，结果如图 5 所示。

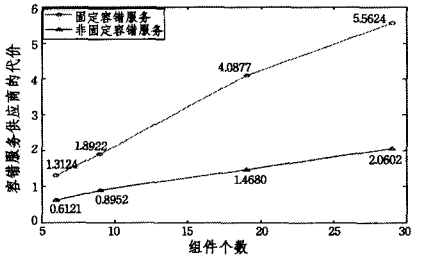

图 5　（资源不足时）容错服务供应商的代价

从图 5 可以看出，非固定容错服务时云容错服务供应商的代价低于固定容错服务时的代价，即非固定容错服务能够更好地满足云容错服务供应商的需求。

综合实验 3—实验 5 可以得出，当云容错服务供应商资源不足时，与固定容错服务相比，本文提出的最佳的容错服务适配方法所生成的容错服务能够使云容错服务供应商用较低的代价为组件提供更优质的服务，这会增加云容错服务供应商的客户量，进而提高云容错服务供应商的收益，即本文提出的容错服务适配方法所生成的容错服务能够更好地满足云容错服务供应商和用户的需求。

结束语　容错服务在云计算容错中起着非常重要的作用。基于可靠性和可靠性误差定义用户容错需求，并为用户提供满足容错需求的非固定容错服务，这不仅能使云容错服务供应商的资源得到灵活应用，从而降低云容错服务供应商的成本，还能降低用户支付的金额，对用户和云容错服务供应商而言具有重要的意义。本文未将时间段之间由于容错服务切换带来的开销考虑在内，未来将在该方面做出更多的研究。

参 考 文 献

[1]　JHAWAR R, PIURI V. Fault Tolerance and Resilience in Cloud Computing Environments (2nd Edition) [M] // Jhawar R, PiuriV. Computer and Information Security Handbook. San-Francisco: Morgan Kaufmann, 2013: 1-28.

（下转第 97 页）

参 考 文 献

[1] PAWGASAME W. Self-Organized TDMA Protocol for Tactical Data Links[D]. Linkoping: Linkoping University, 2011.

[2] BARUAH S, BESTAVROS A. Pinwheel scheduling for fault-tolerant broadcast disks in real-time database systems[C]// Proceeding of the IEEE International Conference on Data Engineering. 1997: 543-551.

[3] QIN Y, ZHANG J, ZHANG T. Effect of TDMA Timeslot Assignment on Traffic Delay[J]. Acta Electronica Sinica, 2009, 30(10): 2277-2283. (in Chinese)
秦勇，张军，张涛. TDMA 时隙分配对业务时延性能的影响分析[J]. 电子学报，2009，30(10)：2277-2283.

[4] LIANG S, MAO Y Q, ZHAO W Q, et al. The Timeslot Assignment Algorithm of Ground-to-Air Data Link[J]. Journal of Force Engineering University (Natural Science Edition), 2005, 6(3): 12-15. (in Chinese)
梁爽，毛玉泉，赵伟光，等. 地空数据链中的时隙分配算法[J]. 空军工程大学学报(自然科学版)，2005，6(3)：12-15.

[5] HUANG K, MAO Y Q, QI Y J. Time-slot Assigning Scheme Used in a New Tactic Network[J]. Journal of Xi'an University of Post and Telecommunications, 2005, 10(3): 23-26. (in Chinese)
黄凯，毛玉泉，戚云军. 一种新型战术通信网的时隙资源分配策略[J]. 西安邮电学院学报，2005，10(3)：23-26.

[6] DONG L, MELHEM R, MOSSE D. Scheduling Algorithms for Dynamic Message Streams with Distance Constraints in TDMA Protocol [C]// Proceedings of EuroMicro Conference on Real-Time Systems. 2000.

[7] MISIC J, CHAN K L, MISIC V B. Performance of Bluetooth Piconets Under E-limited Scheduling[R]. Winnipeg, Manitoba, Canada: University of Manitoba, 2003.

[8] LIU X, HE Z X, ZHOU J L. A Mini-jitter Fixed Timeslot Allocation Algorithm for TDMA Tactical Data Link[J]. Systems Engineering, 2012, 30(6): 90-94. (in Chinese)
刘星，何肇雄，周经纶. TDMA 战术数据链中最小时延抖动的固定时隙分配算法[J]. 系统工程，2012，30(6)：90-94.

[9] CHEN J Y, ZHONG Z D, LIU Q. Time Slot Allocation Algorithm for TDMA Tactical Data Link[J]. Fire Control & Command Control, 2015, 40(8): 97-100. (in Chinese)
陈嘉远，钟章队，刘强. 基于遗传算法的 TDMA 战术数据链时隙分配算法[J]. 火力与指挥控制，2015，40(8)：97-100.

[10] LI X F, DONG Y F, LI J, et al. Data Mining and Knowledge Discovery (second edition)[M]. Beijing: Higher Education Press, 2010: 43. (in Chinese)
李雄飞，董元方，李军，等. 数据挖掘与知识发现(第 2 版)[M]. 北京：高等教育出版社，2010：43.

[11] XIA L Y. Tactical Data Link Technology Research[D]. Xi'an: Northwestern Polytechnical University, 2007. (in Chinese)
夏林英. 战术数据链技术研究[D]. 西安：西北工业大学，2007.

(上接第 67 页)

[2] SUN D W, CHANG G R, MIAO C S, et al. Analyzing, Modeling and Evaluating Dynamic Adaptive Fault Tolerance Strategies in Cloud ComputingEnvironments[J]. Journal of Super Computing, 2013, 66(1): 193-228.

[3] YI H Z, WANG F, ZUO K, et al. Asynchronous Checkpoint/Restart Based on Memory Buffer[J]. Journal of Computer Research and Development, 2014, 51(6): 1229-1239. (in Chinese)
易会战，王锋，左克，等. 基于内存缓存的异步检查点容错技术[J]. 计算机研究与发展，2014，51(6)：1229-1239.

[4] GAO Y, GUPTA S K, WANG Y Z, et al. An Energy-Aware Fault Tolerance Scheduling Framework for Soft Error Resilient Cloud Computing Systems[C]// Design, Automation and Test in Europe Conference and Exhibition. 2014: 1-6.

[5] HAMID B, RADERMACHER A, VANUXEEM P, et al. A Fault-Tolerance Framework for Distributed Component Systems [C]// 34th Euromicro Conference Software Engineering and Advanced Applications. 2008: 84-91.

[6] CHEN S H, CHEN T, WEI H, et al. A Study of Suitable Fault Tolerance Frameworks for an Energy-efficient Storage System [C]// Collaboration Technologies and Systems (CTS). 2015: 220-225.

[7] JHAWAR R, SANTAMBROGIO M. Fault Tolerance Management in Cloud Computing: A System-Level Perspective[J]. IEEE System Journal, 2013, 7(2): 288-297.

[8] NANDI B B, PAUL H S, BANERJEE A, et al. Fault Tolerance as a Service[C]// IEEE 6th International Conference on Cloud Computing. 2013: 446-453.

[9] WU X G. Minimun-cost Based Data Replication Strategy in Cloud Computing Environment[J]. Computer Science, 2014, 41(10): 154-159. (in Chinese)
吴修国. 云计算环境下面向最小成本的数据副本策略[J]. 计算机科学，2014，41(10)：154-159.

[10] YUAN S, GUO Y B, LIU W. Research on voting algorithm in NMR and NVP system [J]. Application Research of Computers, 2008, 25(11): 3463-3467. (in Chinese)
袁顺，郭渊博，刘伟. NMR 及 NVP 系统中表决算法分析与研究[J]. 计算机应用研究，2008，25(11)：3463-3467.

[11] DENG W P. Research on checkpointing in mobile computing environment and modeling with petri nets[D]. Harbin: Harbin Institute of Technology, 2008. (in Chinese)
邓文平. 移动计算环境下检查点技术研究与 Petri 网建模[D]. 哈尔滨：哈尔滨工业大学，2008.

[12] ZHAO L H. Search of Cloud Task Allocation Strategy Based on Cloudsi[D]. Hohhot: Inner Mongolia University of Technology, 2013. (in Chinese)
赵立慧. 基于 CLOUDSIM 平台的云任务分配策略研究[D]. 呼和浩特：内蒙古工业大学，2013.

[13] 赵东方. 数学模型与计算[M]. 北京：科学出版社，2007：249-274.

[14] REN J Z, TAN S Y, DONG L C, et al. Lingo optimization software and its application on optimization of chemical process[J]. Computers and Applied Chemistry, 2010, 27(7): 975-978. (in Chinese)
任竞争，谭世语，董立春，等. LINGO 及其在化工过程优化中的应用[J]. 计算机应用化学，2010，27(7)：975-978.

[15] ZHENG Z B, LYU M R. Fault Tolerance Management in Cloud Computing: Selecting an optimal fault tolerance strategy for reliable service-oriented system with local and global constraints [J]. IEEE Transactions on Computers, 2015, 64(1): 219-232.