一种新的云存储服务模型研究 *

刘金芝, 余 丹, 朱率率
(江苏技术师范学院, 江苏 常州 213000)

摘 要: 提出了一种基于云存储的可生存存储服务模型,对模型架构进行了详细设计和构造,并采用了一种存储服务模型可用性的度量方法(资源分配失效概率模型)进行了分析。通过分析可得,提出的可生存云存储服务模型能够满足较高的可生存性的需要。

关键词: 可生存性; 云存储; 服务模型; 云计算
中图分类号: TP393 **文献标志码**: A **文章编号**: 1001-3695(2011)05-1869-04
doi:10.3969/j.issn.1001-3695.2011.05.079

Research on new cloud storage service model

LIU Jin-zhi, YU Dan, ZHU Shuai-shuai
(*Jiangsu Teachers University of Technology*, *Changzhou Jiangsu* 213000, *China*)

Abstract: This paper proposed a new survivable cloud storage service model, and designed the structure of the model and constructed in detail. Then adopted a storage service survivability evaluation method (resource allocation failure probability model) based on probability models to analyze the service model. The fully evaluation justifies that the cloud storage service model can satisfy survivability.

Key words: survivability; cloud storage; service model; cloud computing

0 引言

可生存性的概念由 Barnes 等人[1] 于 1993 年首先提出。人们普遍认可的可生存性的定义为:在出现故障、意外事件或针对系统的攻击时,系统所具有的及时完成任务的能力[2]。文献[3]中提出了一种通用的可生存性的定义。系统可生存技术的研究建立在相关研究领域,如安全性、容错、可靠性和可测性等研究基础之上,并引入了一些新的概念和规则。传统的计算机安全侧重于保护信息的机密性,而可生存性更注重保护信息的可用性和重要服务的持续性。可生存网络系统的两个基本特征是:a)出现攻击时保护系统的基本服务并及时地全面恢复所有服务;b)在网络体系结构动态变化的过程中保证系统的可生存性。

生存性参考值可以用故障发生前业务量与故障发生后业务量的比值来体现。对于生存性来说,显然要求该值越大越好,最好是故障对于系统的主要服务没有影响。这一点通过设计来实现。为了达到可生存性,必须在系统中引入冗余,但过多的冗余会降低系统工作效率,可生存性是性能(包括安全性、可用性、可靠性等)与效率之间的折中,因此在实际应用中必须有一种基础性平台,可以提供合适的数据冗余度和方便的服务弹性,"云"便是一个优秀的候选平台。

"云"是一些可以自我维护和管理的虚拟计算资源,包括计算服务器、存储服务器、带宽资源等。云计算是分布式处理、并行处理、网格计算、网络存储和大型数据中心的进一步发展和商业实现,是基于互联网的超级计算模式。云计算是以各种服务的形式而存在的,云计算的一系列特点使它成为可生存性研究最有潜力的平台。因此,提供动态的数据服务成为云计算越来越重要的应用之一。云存储服务模型是以云计算为基础架构,通过集群应用、网格技术或分布式文件系统等功能,将网络中大量各种不同类型的存储设备通过应用软件集合起来协同工作,共同对外提供数据存储和业务访问功能的一个系统,如 Amazon 的 EC2[4]、IBM 的 blue cloud[5] 等。

本文将云抽象为目标网络区域内一系列资源的优化配置,每一种配置都可以支持来自内部或外界的一种或多种服务请求,并以可生存性的两个基本特征为原则,提出一种云存储服务模型,通过持续性地构造和优化组合以达到服务可生存性的目地。

1 相关工作

目前的可生存技术研究主要是从系统设计[2,3] 和系统框架优化[6] 的角度来保证系统的可生存性,一般开销较大且可行性较差。可生存性重点在增强应急响应和故障恢复方面的能力,让系统对入侵和攻击具有弹性,只要采取措施保障关键服务能够持续运行,系统就能够在攻击中生存下来。

PASIS[7] 是 Carnegie Mellon University 的研究人员设计的一种存储系统模型,其使用秘密共享方案来实现可生存性。

收稿日期: 2010-10-03; **修回日期**: 2010-11-23 **基金项目**: 青年科研基金资助项目(yy09037)
作者简介: 刘金芝(1980-),女,讲师,主要研究方向为计算机存储、信息安全等(liujz@jstu.edu.cn);余丹(1986-),男,硕士,主要研究方向为计算机安全;朱率率(1985-),男,硕士,主要研究方向为计算机存储、信息安全.

Carnegie Mellon 的 Jay 等人[8]还提出了设计可生存的存储系统的协议族方法,这种方法将系统可以容忍何种错误的问题由系统的设计阶段转移到了数据创建阶段。协议族通过简单地改变存储节点的数量和一些读写门限来提供对于不同故障模式的处理。一个协议族使给定的存储结构能够用于各种错误模式中,并允许故障数目与数据无关。文献[9]中提出了分布式存储系统的代理系统,每个代理提供分布式存储节点中基本的存储服务,如保存、调用、更新等,并提供用户接口。这种代理系统可以保证信息的机密性、可靠性及可生存性,其中可生存性也是通过将数据秘密共享分散保存而实现的。

目前,云计算理论迅速发展,但云存储服务模型的理论研究仍无人涉足。在云计算中,资源的重新分配是发挥云平台作用的关键技术。其中比较有代表性的资源重配算法主要是以完成时间[7]、资源的吞吐量、使用率[10]等作为重配目标,都没有直接的云存储服务的系统性研究,也没有考虑云存储服务的可生存性问题。同时,在可生存技术中,资源配制机制的引入又需要考虑一定的限制。a)资源配制算法要简捷高效,服务响应与故障恢复时间要尽可能小;b)服务要有优先级机制,以保证重要服务得到响应;c)充分保证云中资源的平衡性和稳定性,不能因提高某些服务的可用性而导致平台的不稳定。

2　可生存的云存储服务模型要素及相关结论

2.1　服务模型要素描述

可生存的存储服务可表示为一个四元组 $\Gamma = \langle E, C, S, F \rangle$。其中:$E$ 为系统运行环境的描述;C 为对当前服务提供支持的资源配置的描述;S 为系统对用户提供的服务类型集的描述;F 是故障模型,包括故障类型及处理方法。具有容忍入侵特性和灾难恢复功能的系统结构中通常会涉及一个请求服务的客户集和一个以冗余的方式为客户提供服务的服务器集,而本文将二者合并为一个可操作的云滴集合。

为了发挥云存储配置灵活、超大模型和低成本等诸多优势,可生存的存储服务四元组在云计算的框架内可以作如下的实例化。

1)运行环境 E

E 为对于存储系统所处环境的说明。根据互联网上存储介质的特性,云定义假设所有存储服务设备构成正态云 $c(X)$,每个存储介质构成一个云滴 $x, x \in X$,每个云滴是由一系列资源组成的集合 $x_i = \{R_1, R_2, \cdots, R_n\}$。$E$ 中的每一个存储节点和用户都可以根据自身相应的属性映射为云中的一个云滴。另外,根据各个网络节点的计算能力、存储能力、平台相同与否、是否在网络时间上同步与异步、节点运行的环境是否安全等都可以抽象为节点上资源数量和种类的不同组合。

2)资源配置描述 C

当从云的某个位置发起一个服务 s 时,对 s 的支持可以在短时间内细化为一系列资源的配置,并由调度算法分配到云的其他部位,从而对 s 快速响应。资源组合 $R_1, R_2, \cdots, R_m$ 在配置规则 Φ 控制下,以概率 p 对服务 s 提供支持,即资源组合到服务集的映射,其有效的概率为 p,其余映射可理解为因资源调度失败而中止的服务响应。

$$\Phi(R_1, R_2, \cdots, R_m) \xrightarrow{p} s$$

3)服务类型描述 S

存储系统中的服务包括数据的保存、调用、更新、删除等,本文将所有服务简化为读取和更新两种。读取服务过程即某些云滴或来自外部的对数据的下载请求,然后通过云计算中数据处理算法将相关数据恢复。更新服务是指选择优化的云滴为目标存储节点,并利用恰当的数据分散算法,将数据分割保存到各节点中或将目标数据更新至最新版本。一个服务的描述可以用一个结构 Ψ 来表示。$\Psi\{O, M, C\}$,其中:O 为优化存储策略,M 为数据拆分与组合算法,C 为资源配置算法。

4)故障模型 F

在可生存性研究中,人们通常将系统中的故障分为两种:损毁故障(crash)和 Byzantine 故障,或者良性故障和恶性故障。损毁故障指一般的硬件或通信线路故障,导致数据不能正常传送。而 Byzantine 故障通常指随机错误,即发生故障的服务器的行为是随机的,这可以理解为软件故障,也可以包括由于攻击者入侵而导致对正常服务的破坏。

因此,云存储服务故障可以分为两类:因资源配置引起的服务不能进行和因随机或人为因素引起的服务不一致。后者主要通过服务完整性验证完成,不详细讨论。前者可以分为云滴资源配置故障Ⅰ和云滴集资源配置故障Ⅱ。当对某个云滴 x 的所有资源请求均失败,即 $\Phi(R_1, R_2, \cdots, R_m) = \perp$ 时发生故障Ⅰ,资源配置算法将自动请求下一个可用的云滴;当在当前的网络拓扑中请求资源时,没有足够的云滴做出响应,则会产生云滴集资源配置故障Ⅱ,服务将在备选的云滴集中请求资源,直到用户的计时器超时或系统自动选择另一时刻进行资源的重配置。

故Ⅰ类故障的判定条件是

$$\Phi_x(R_1, R_2, \cdots, R_m) = \perp$$

Ⅱ类故障的判定条件是

$$\Phi_C(R_1) \cap \Phi_C(R_2) \cap \cdots \cap \Phi_C(R_m) = \perp$$

Ⅰ类故障是构成Ⅱ类故障的因素,但不是包含关系。云内Ⅰ类故障的发生是以不可忽略的概率存在的,Ⅱ类故障则是以极小的概率发生的。

2.2　相关结论

1)云存储服务存在性

假设在网络中大量的计算与存储部件其资源可用的确定度具有稳定倾向性,其分布构成云 $C(Ex, En, He)$,并假设存在映射 $\pi: C \rightarrow \Pi$ 满足正态云算子的约束条件,使 C 变换为用数值表示的云滴集合。

定理 1　假定云滴 $\mu(R_1, R_2, \cdots, R_n)$ 在时间 $[0, t]$ 内的资源可以被成功请求的概率为 p_i,对于服务请求 Ψ 在时间 $|t|$ 内成功返回资源配置 $c(R_1, R_2, \cdots, R_m)$ 的概率为 P,则存在 ε 使 $P > \varepsilon$。

证明　根据古典概率模型即可得证。过程略。

定理 1 说明当满足一定的条件时,支持目标服务的云滴集合是以一定概率存在的。当网络区域一定时,在存储服务四元组 $\Gamma = \langle E, C, S, F \rangle$ 中唯一变动的因子是 S,只有通过运用合适的服务类型描述才能使 ε 达到满意的程度。

2)保存服务的优化

定理 2　对于给定的网络区域 E,存在算法 A 可以发现优化的 Ψ,使 P 取得最大值。

证明　即证存在算法 A,发现满足下列条件的目标变量 P 的最大值问题。

$$\begin{cases} c(R_1, R_2, \cdots, R_m) \xrightarrow{\Psi_0, P} s \\ P = 1 - P(\mu(R_1, R_2, \cdots, R_n) = \perp) \\ \Psi(O, M, C) \mid_{\Psi_0} \end{cases} \tag{1}$$

求解该问题可归结为从初始服务描述 Ψ_1 在进化算法 A 的作用下,经过遗传和变异,最终收敛为目标 Ψ_0。文献[11]中的 CBEA 算法即可作为高效、高精度的候选算法,以式(1)作为约束条件即可发现优化的 Ψ_0。

3　可生存的云存储服务模型构造

3.1　保存服务 DS

保存服务的关键在于优化存储策略 O 和数据拆分与组合算法 M 的选择。

存储策略是指选择哪些存储节点来保存数据份额。假设当前用户采用数据分散算法 D 将某个数据单元分为 n 个份额。在云中可观测到 N 个可用的云服务器节点,一般情况下由于数据的地理位置、重要程度以及拥有者不同,以及出于效率上的考虑,不可能利用所有这些节点来保存这 n 个份额,即对某一个数据单元来说,$n \leqslant N$。因此,在保存某个数据时,要考虑如何在 N 个节点中选择 n 个来保存相应的份额。文献[12]中提出了四种常用的存储策略:a)随机地选择 n 个节点;b)将一个份额保存在指定节点,其他 $n-1$ 个节点随机选择;c)检查每个节点可利用的存储空间,根据负载均衡的原则来选取;d)所有的数据都选择同样的 n 个节点。文献[13]中通过对生存率和空间使用率的实验,得出 a)是最好的存储策略。

数据拆分与组合方案则是一个一到多映射,表示由原始数据计算 n 个份额的算法。但是,数据的分散保存不应局限于份额复制或纠删编码方案等,而是要根据数据的重要性、特点,以及侧重于支持的性能来选择多种不同的方案。

当由式(1)成功地返回优化资源配置描述时,服务请求将得到正确的响应,数据份额将按照资源配置进行分发保存。

3.2　恢复服务 RS

数据的恢复是一个多个数据份额 d_i 到原始数据 D 的多对一映射。数据恢复时随机选择 k 个可能提供资源的云滴 $x_1, x_2, \cdots, x_k$,发送 RS 服务请求,节点验证用户身份后,发送 k 个份额,用户按映射关系 θ 计算:

$$\theta: D \times D \times \cdots \times D \mapsto D$$

$$\theta(d_{i_1}, d_{i_2}, \cdots, d_{i_k}) = m'$$

若 k 个节点都无故障,数据恢复成功,消息内容与原始消息一致,即 $m' = m$;否则,m' 为语义上无意义的信息,k 个节点中必存在故障节点,按 I 类故障模型处理,重新选择相应资源的另外云滴发送请求。若在一定时间限度内收不到期望传来的数据,则可认为该云滴集合发生了 Ⅱ 类故障,并作相应的处理。

3.3　更新服务 US

本文认为更新服务 US 包含除创建和读取之外的所有服务内容,对数据进行更新时必须更新所有的 n 个份额。因此更新服务包括以下步骤:用户向 k 个节点发送读请求;恢复数据(RS);修改完毕后计算新的数据份额;向 n 个节点发送修改请求,完成修改。

3.4　服务响应过程

如图 1 所示,服务生命期内主要经过以下几个过程:

a)发起服务进程。若云滴 A 需要某个数据服务,则 A 将发起一个包含服务请求的初始进程,并交由云中的云滴集合维护。

b)服务类型产生。在云滴集合初始化完毕后,将产生由优化存储策略、数据拆分与组合算法,以及资源配置算法组成的服务类型描述。它是由多个算法标记及相应参数组成的描述符,可以通过云滴集合的演变而得到优化。

c)资源配置。正常的云滴将分配部分资源给当前的服务请求,云滴集合不断地演变,直到所有需求的资源得到满足。这些资源在统一的接口控制下返回给云滴 A。

d)服务响应。当云对服务可以提供支持以后,服务发起方与优化的云滴集合进行数据交互,完成服务主体内容。

e)返回服务结果。当服务正常结束时,返回给原始云滴一个确认信息,否则,返回出错信息。如图 2 所示,云滴 A 可以选择放弃服务,也可以选择进入服务故障处理模式,重新发起服务。

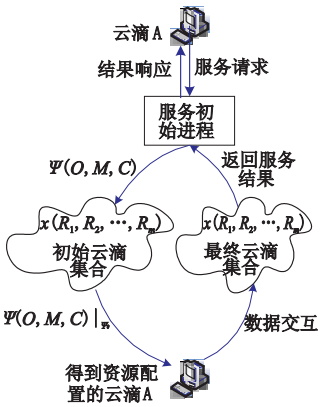

图1　服务生命周期

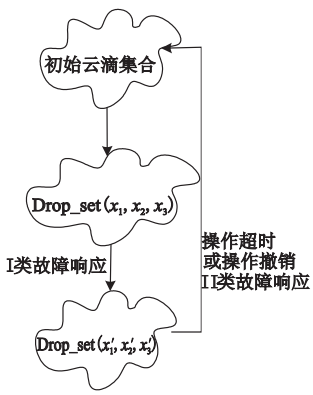

图2　服务故障响应模式

服务生命周期进行过程中存在两类故障,并分别有两种不同的响应模式。

1) Ⅰ 类故障响应

由云响应来自某个云端的数据服务请求,如当要创建或修改数据单元时,若某个云滴对资源的支持失败则产生 I 类故障响应。

2) Ⅱ类故障响应

该类服务响应主要针对来自云滴集合的故障处理请求。当在某个区域有大量 I 类故障响应失败时,当前服务请求必然失败,转入对 II 类故障的响应。

4 服务的可生存性评估

服务的可生存性评估是存储系统研究的热点,目前有多种模型,如文献[14]分别从简单概率模型和基于失效概率密度函数的模型两种方法分析了存储系统的可靠性。为了分析了本文服务模型的可用性,采用了资源分配失效概率模型。

一个失效的服务请求其实质是请求的资源没有得到支持。在资源分配失效概率模型中,假定云滴资源状态服从简单概率模型,即假设这些概率是先验地测定的,并具有固定的值,且各云滴之间相互独立。

设对于由 $\Psi(O,M,C)$ 选定的云滴集合为 $C(d_1,d_2,\cdots,d_u)$,$u>m$,且 Ψ 中对资源 R_j 需求的概率为 $p(R_j)$,d_i 可提供资源 R 的概率为 $p(d_1,R)$,分别服从正态分布 $N_1(\mu_1,\sigma_1^2)$,$N_2(\mu_2,\sigma_2^2)$,C 可提供资源 R 的概率为 $P(R)$。

发生 Ⅰ 类故障的可能性可表示为

$$P_1(d_i)=p(R_1)[1-p(d_i,R_1)]+p(R_2)[1-p(d_i,R_2)]=\sum_{j=1}^{m}\{p(R_j)[1-p(d_i,R_j)]\}$$

其中:$P_1(d_i)$ 满足

$$1-\sum_{j=1}^{u}p(d_i,R_j)\leqslant P_1(d_i)\leqslant\sum_{j=1}^{u}p(d_i,R_j)$$

又因为

$$P(R_i)=\sum_{j=1}^{u}p(d_j,R_i)$$

故发生Ⅱ类故障的概率为

$$P_{\mathrm{II}}(C)=1-\prod_{i=1}^{m}P(R_i)=1-\prod_{i=1}^{m}\sum_{j=1}^{u}P(d_j,R_i)$$

例:云滴 $C(d_1,d_2,\cdots,d_{50})$ 中的元素 d_i 对某三种资源(R_1,R_2,R_3)的支持概率 $p(d_i,R_j)$ 服从分布 $N_1=N_2=N(\mu,\sigma^2)$,$\mu=0.5$,$\sigma^2=1$,当前服务 φ 对某种资源存在需求的概率和单个云滴对该资源的支持概率由该正态分布随机产生,则三次取样计算得表 1。

表 1　三次随机的资源失效概率计算

概率	三次取样		
$P_{\mathrm{I}}(d_1)$	0.132	0.118	0.151
$P_{\mathrm{II}}(C)$	0.002	0.001	0.005

经验证,当 $\mu>0.7$,$\sigma^2<1$ 且 $n>50$ 时,$P_{\mathrm{II}}(C)<0.00001$,故服务模式的可用性较高。

5 结束语

随着数据量的指数级增长和互联网服务体系的完善,面向服务可用性的云存储技术必将取代传统大容量分布式存储系统成为网络存储的主流。本文从资源分配的角度设计了基于云的存储服务模型,并详细探讨了模型的响应过程,最后对该模型的可用性作了评估,并认为其可用性可以达到较高的水平。

参考文献:

[1] BARNES H A, NEUMANN P G. Survivable computer-communication systems: the problem and working group recommendations, VAL-CE-TR-92-22 (revision 1)[R]. [S. l.]: US Army Research Laboratory, 1993.

[2] ELLISON R J, FISHER D A, LINGER R C, *et al*. Survivable network systems: an emerging discipline, CMU/SEI-97-TR-013[R]. Pittsburgh: Carnegie Mellon University, 1997.

[3] KNIGHT J C, STRUNK E A, SULLIVAN K J. Towards a rigorous definition of information system survivability[C]//Proc of DARPA Information Survivability Conference and Exposition. 2003.

[4] DEJUN J, PIERRE G, CHI C H. EC2 performance analysis for resource provisioning of service-oriented applications [C]//Proc of Workshop on Non-Functional Properties and SLA Management in Service-Oriented Computing. 2009.

[5] IBM blue cloud project[EB/OL]. (2008-06). http://www-03.ibm.com/press/us/en/pressrelease/22613.wss/.

[6] CECCARELLI A, GRONBAEK J, MONTECCHI L, *et al*. Towards a framework for self-adaptive reliable network services in highly-uncertain environments[C]//Proc of the 13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops. 2010: 184-193.

[7] WYLIE J, BIGRIGG M W, STRUNK J D, *et al*. Survivable information storage systems[J]. IEEE Computer, 2000, 33(8): 61-68.

[8] WYLIE J J, GOODSON G R, GANGER G G, *et al*. A protocol family approach to survivable storage infrastructures[C]//Proc of the 2nd Bertinoro Workshop on Future Directions in Distributed Computing. 2004: 23-25.

[9] LEGRAND I, NEWMAN H, VOICU R, *et al*. MonALISA: an agent based dynamic service system to monitor, control and optimize distributed systems[J]. Computer Physics Communications, 2009, 180(12): 2472-2498.

[10] ATENIESE G, FU K, GREEN M, *et al*. Improved proxy re-encryption schemes with applications to secure distributed storage[C]//Proc of the 12th Annual Network and Distributed System Security Symposium. 2005: 83-107.

[11] ZHANG G W, HE R, LIU Y, *et al*. An evolutionary algorithm based on cloud model[J]. Chinese Journal of Computers, 2008, 31(7): 1082-1091.

[12] HAYASHI D, MIYMOTO T, DOI S, *et al*. Agents for autonomous distributed secret sharing storage system[C]//Proc of the 2002 International Conference on Circuit/Systems Computers and Communications. 2002: 482-485.

[13] ZHU S S, YANG X Y, ZHANG W. Schedule algorithm based on data relativity[J]. Computer Engineering, 2010, 36(10): 80-87.

[14] ZHANG W, MA J F, YANG X Y. Reliability of distributed storage systems[J]. Journal of Xidian University, 2009, 36(3): 480-485.