一种新的云存储数据容错存储方式检验方法

纪 倩[①] 杨 超[*①] 赵文红[②] 张俊伟[①]
[①](西安电子科技大学计算机学院 西安 710071)
[②](嘉兴学院南湖学院 嘉兴 314001)

摘 要：云存储中，防止数据丢失的关键是实现文件容错。然而，云存储服务商可能没有提供承诺的容错水平，导致用户蒙受数据丢失和经济损失的双重风险。现有云存储数据容错存储方式检验方法存在服务器预读取欺骗攻击，并且效率低、实用性差，不能达到在一定概率范围内，快速、轻量级地检测出犯规的服务器行为的要求。针对上述问题，该文利用磁盘顺序存取和随机存取的差异性设计了一种远程数据容错存储方式检验方法 —— 随机与顺序访问时间差异化(DRST)方法，其原理是文件块被分散地放在不同磁盘上，读取一个磁盘上顺序存储的文件块比随机读取不同磁盘上的文件块所需的响应时间短。最后，对所提方法进行了严格的理论证明和深入的性能分析，结果表明，所提方法能够快速检验出服务器是否为用户提供了其承诺的容错水平，并且比现有方案更安全，更高效。

关键词：云存储；云文件安全；数据容错能力；随机存取；顺序存取
中图分类号： TP309 **文献标识码**： A **文章编号**：1009-5896(2016)10-2640-07
DOI: 10.11999/JEIT151344

New Method for Checking the Data Stored with Fault Tolerance in Cloud

JI Qian[①] YANG Chao[①] ZHAO Wenhong[②] ZHANG Junwei[①]
[①](*School of Computer Science and Technology*, *Xidian University*, *Xi'an* 710071, *China*)
[②](*Nanhu College*, *Jiaxing University*, *Jiaxing* 314001, *China*)

Abstract: Implementation of file fault tolerance is the key for preventing data loss in cloud. However, cloud storage service providers may not offer the committed level, which results in that users may suffer data loss and economic loss. Existing inspection algorithms of testing of data fault tolerance in cloud have disadvantages such as spoofing attack of pre-fetch, low efficiency and poor practicality, which can not detect the foul behavior of cloud storage providers within a certain probability. To deal with the above problems, utilizing the difference of sequential access and random access, a remote testing algorithm of data fault tolerance in cloud named (Difference of Random and Sequential access Time) (DRST) is designed. The core idea is that the time of reading blocks of a file stored in order on a disk is much shorter than reading blocks of a file stored random on different disks. A strict theoretical proof and a in-depth performance analysis to the proposed scheme are carried out. The results show that the proposed scheme can accurately detect whether the cloud storage provider supplies clients with the committed level of fault tolerance.What's more, the proposed scheme is much more efficient than the existing ones.

Key words: Cloud storage; Cloud file security; Data fault tolerance; Random access; Sequential access

1 引言

云存储中安全问题是用户最大的质疑和担心。用户将数据放到云端，就丧失了对数据的绝对控制权。实际的云存储环境下，数据主要分布存储在数据中心上，存储其上的数据规模往往达到 PB 级甚至 EB 级，数据失效成了一种常态行为，极大地限制了云存储的应用和推广[1]。因此，云存储系统的数据容错十分重要，直接关系到整个系统的可用性。

传统的存储服务商通常使用几个 9 —— 如 99.99%, 99.9999999%形式的 SLA(服务等级协议)向客户提供不同级别的容错能力。现今的云存储服务商往往通过采用基于纠错码的冗余机制来提供数据的容错保证服务，即将一份增添了纠错码的文件分布存储到一个数据中心的多个磁盘上，使其可以抵抗一定范围内的硬件故障，如，将增添了冗余码的

收稿日期：2015-12-01；改回日期：2016-07-12；网络出版：2016-08-26
*通信作者：杨超 chaoyang@mail.xidian.edu.cn
基金项目：国家自然科学基金青年基金(61303219)，国家自然科学基金(61672415)，中央高校基本科研业务费(JB140303)
Foundation Items: The National Natural Science Youth Foundation of China (61303219), The National Natural Science Foundation of China (61672415), The Fundamental Research Funds for the Central Universities (JB140303)

文件存储在 4 个磁盘上便可以抵抗至多 3 个磁盘的崩溃[2]。实际上由于云存储服务商向用户隐藏了文件块存储的布局信息，用户无法真正了解他们的文件到底享受着什么程度的容错能力[3,4]。因此，对于用户来说，及时、可靠和便捷地远程验证云存储服务商是否达到所声称的“抗多磁盘崩溃”的数据容错水平是非常必要和紧迫的[5]，而且这是新型云存储大规模应用和推广前亟需解决的问题。

早先，文献[6,7]采用纠错码与随机抽样相结合的请求响应式协议进行了数据拥有证明(PDPs)[8]和可恢复性证明(PORs)[6,9,10]，用于远程验证存储在云端的文件的完整性。发表于 2011 年 ACM CCS 的文献[3]针对远程验证云服务供应商是否将文件均衡存储在多个磁盘设备的问题，首次提出了一种实用性评估方法 RAFT(Remote Assessment of Fault Tolerance)。其核心思路是用户与服务商事先商量，就文件布局策略达成一致的前提下，将文件分配到多个磁盘，磁盘数目越多，文件块分布越均衡，服务器并行读取文件块的速度越快，若服务商使用较少的磁盘或文件分布不均衡，会使响应时间变长，以此来判断服务商没有将文件存储到承诺的多个磁盘上。但 RAFT 方案在安全性及实用效率方面存在如下 2 个问题：

(1)服务器预读取的欺骗攻击：　若服务器没有按照要求根据上一步访问的文件块内容做哈希产生下一步要读取的块序号，而是把预先读取的块作为随机产生的请求块返回，从而使响应时间降低，用户将无法根据响应时间察觉服务商的作弊行为，就会被云服务供应商所欺骗，数据被置于极大的丢失风险中。。

(2)效率低实用性差：　该方案本质上只利用了数据并行读取和串行读取的时间差，该时间差比较小，尤其是在高性能磁盘情况下，要区分诚实服务商与狡诈服务商，需要进行很多步请求，才能使总的时间差比较明显，性能较低。

文献[11,12]也针对远程验证云服务供应商是否将文件均衡存储在多个磁盘设备的问题提出了新方案，其主要特点是文件被随机存储于任意磁盘上，要求云服务供应商针对不同程度的容错冗余程度发布官方的响应时间。但实际上，由于文件存储布局是随机的，即使存储文件的冗余程度相同，其响应时间也可能不尽相同，因此要求服务商提供官方的响应时间不合理，而且文献[12,13]缺少相关的性能分析、评估及对比。

针对现有方案的不足，本文提出了随机与顺序访问时间差异化 —— DRST(Difference of Random and Sequential access Time)方法。该方法的主要思想是，假设服务器存储文件时只存逻辑文件块与所在磁盘的对应关系，不会记录或者优化每个磁盘上存储的文件块的顺序，这样服务器不能提前判断请求的文件块在哪个位置，所以，针对每一次请求，服务器必须查询逻辑文件块与所在磁盘的对应关系表，查表是不可避免的，这样可以显著区分诚实和欺骗的服务提供商行为。由于磁盘随机读取和顺序读取的时间差比较大，该方法可以用较少的步数，快速、轻量级地检测出犯规的服务器行为，从而判断服务器是否按容错承诺将文件存储在多个磁盘上。

2　DRST 方法设计

2.1 DRST 方法设计思想

服务供应商可能是廉价且懒惰的[3]，即存储较少的冗余在较少的磁盘上，这些“狡诈服务供应商”可能提供不合格的容错，但不会恶意攻击故障磁盘的安全漏洞。针对这个问题，新方案的设计思路是文件块被分散地放到不同的磁盘上，读取一个磁盘上顺序存储的文件块比随机读取不同磁盘上的文件块所需的响应时间短。即基于磁盘顺序存取和随机存取的差异性，我们设计了一种远程数据容错存储方式检验方法 DRST，该方案不仅可以检测服务商是否将文件分布存储在多个磁盘上，还可以证明文件分布是否均衡。

DRST 方法特点：(1)在服务器端按照规定的存储策略存储文件的前提下，由用户端提出请求向服务器端说明每一步要读取的文件块序号，服务器并不知道用户端下一步要读取哪一块，无法预取来迷惑用户，可防止服务器将预取的文件块返回；(2)每步只请求一个文件块，多步请求的文件块实际是顺序放在一个磁盘上的，而服务器事先不知道也无法预测，若服务商不诚实没有遵守约定，而使用较少磁盘的话，文件块可能分布在不同磁盘上，这样读取的文件块的存储位置呈现一定的随机性，寻道次数增多，用户则可通过变长的响应时间辨别服务质量。由于磁盘随机读取和顺序读取的时间差相对较长，使用较少的步数，便可快速、轻量级地检测出犯规的服务器行为。

2.2 DRST 方法详细设计

为了更正式地定义我们的系统，我们首先引入一些符号。把 l 作为一个安全参数，f_i 表示文件 F 的第 i 个块，其中 $i \in \{1,2,\cdots,|F|\}$。我们用 C 表示用户端，用 S 代表云服务器端，C 与 S 的主要交互过程如图 1。

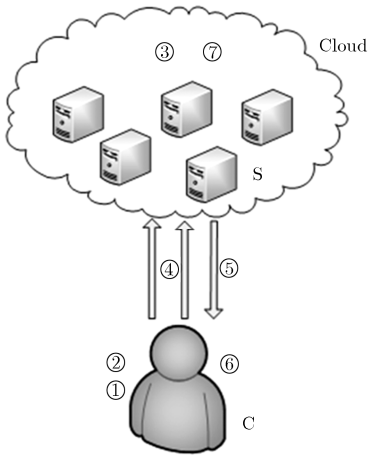

图 1 用户端与云服务商交互图

具体步骤如下：

步骤 1 $\text{Keygen}(1^l) \xrightarrow{R} k$ ：密钥生成函数，输出密钥 k(由 C 完成)。

步骤 2 $\text{Encode}(k, F=\{f_i\}_{i=1}^m, t, c) \to G=\{g_i\}_{i=1}^n$: C 对具有 m 个文件块的文件 $F=\{f_i\}_{i=1}^m$ 增加冗余码，需要输入能够承受的崩溃磁盘数目 t 和总磁盘数目 c，输出经过编码的文件 $G=\{g_i\}_{i=1}^n$，m 块文件扩增为 n 块，$n=mc/(c-t)$ 。

步骤 3 $\text{Map}(n,t,c) \to \{C_j\}_{j=1}^c$:S 将 n 块文件存在 c 个磁盘，输出 $C_j=\{n \mid n \bmod c = j\}$，$C_j \subseteq \{1, 2,\cdots,n\}$，表示存储在第 j 个磁盘上的文件块序号集合，其中 $j \in \{1,2,\cdots,c\}$。如果输出不是 $\perp$，这样的布局就可以承受 t 个磁盘的崩溃。

步骤 4 $\text{Challenge}(n,G,t,c) \to Q$：输入编码后的文件块数目 n，编码文件 G，容错水平 t 和磁盘数目 c，C 指定读取第 j 个磁盘上的文件块，其中 $j \in \{1,2,\cdots,c\}$，按照从小到大的顺序依次请求 C_j 中的序号对应的文件块，返回一个文件块后请求下一个文件块，发出 q 次请求。

步骤 5 $\text{Response}(Q) \to (R,T)$:S 使用存储在磁盘中已编码的文件块来响应用户端发出的请求。C 记录从发送一个请求到接收到来自 S 的响应 R 所经过的响应时间 T 。

步骤 6 $\text{Verify}(G,Q,R,T) \to b \in \{0,1\}$:C 验证文件块是否与原文件一致，响应时间是否在可接受的时间间隔内(用户端存有文件的副本，用于验证)，输出 1 表示 C 成功地验证了 S 是按照规定存储文件的，达到了承诺的容错水平；相反，0 表示 S 有犯规行为。

步骤 7 $\text{Reconstruct}(k,r,\{g_i^*\}_{i=1}^r) \to F^*=\{f_i^*\}_{i=1}^m$:S 根据 r 个已编码的文件块恢复原文件 F (m 个文件块)。

3 DRST 方法安全性分析与性能评估

3.1 DRST 方法安全性分析

本文方案的设计目标是通过多步请求来识别在容错级别上欺骗用户的存储服务商，具体来讲，用户与存储服务器事先约定好存储布局，然后通过随机多步的挑战响应交互时间差来判别存储服务商是否按照用户要求对数据进行了容错存储。本文方案使得存储服务器在约定好的容错级别和存储布局的前提下，不能通过预先读取等手段来欺骗用户，除非存储服务器可以提前猜测到用户随机的数据块请求顺序。实际上，我们有定理 1 如下：

定理 1 对于所提出的 DRST 方法，如果随机数据请求的次数为 q，存储服务器能欺骗用户通过容错级别验证的复杂度远远小于猜中随机数 $r \in \{0,1\}^q$ 的概率，即 $\Pr[\text{Guess}_s] << 1/2^q$ 。

证明 假设用户在 DRST 方法中对存储服务器进行 q 次随机数据块的请求，远程的存储服务器能够在没有按照容错级别要求存储用户数据的前提下通过 DRST 方法的挑战应答响应，使得用户不能觉察响应时间较长，即用户返回验证正确，进而欺骗用户已经达到承诺的容错水平。这需要远程存储服务器对用户的 q 次随机数据请求的响应时间(包括磁盘寻道和数据读取时间) T_q^r 与同一个磁盘上顺序存放数据块请求的响应时间 T_q^s 非常接近，即 $T_q^r - T_q^s = \varepsilon$，其中 ε 是个可以忽略的值。即，需要远程存储服务器读取随机磁盘位置数据的时间与读取顺序磁盘位置数据的时间几乎一样。为了达到这样的要求，要么磁盘寻道不需要时间，要么服务器能提前预先读取数据块。当磁盘寻道不需要时间为真时，这与现有物理磁盘寻道时间不为零相互矛盾；当服务器能提前预先读取数据块为真时，这要求服务器能正确猜测用户 q 次的随机位置的数据请求，在这种情况下，假设一个磁盘可存放数据的位置共有 L 处，则服务器正确猜测 q 次用户随机位置的数据请求的概率为 $\Pr[\text{Guess}_s] = 1/[L(L-1)(L-2)\cdots(L-q+1)]$；因为本文方案的查询次数 q 一般小于 100，而磁盘的数据存储位置 L 一般要比 100 大几个数量级，所以 $L >> q$，即 $L-q >> 0$，则 $\Pr[\text{Guess}_s] << 1/2^q$；进一步而言，当 $q=128$ 时，正确猜测 q 次用户随机位置的数据请求的概率为 $\Pr[\text{Guess}_s]$ 比直接正确猜测一个 128 比特位的加密密钥还要小。

证毕

3.2 DRST 方法性能的理论分析

根据 DRST 方法，诚实服务商读取文件块时间：

$$T_h = t_l + t_s + q \times t_r \tag{1}$$

狡诈服务商读取文件块时间：

$$T_t = t_l + q \times t_s + q \times t_r \tag{2}$$

时间差：

$$\Delta t_D = (q-1) \times t_s \tag{3}$$

其中 q 表示请求次数，即要读取的文件块数，t_l 是网络延迟时间，t_s 是磁盘寻道时间，其中 t_r 是读取文件块的时间。

RAFT 方法[3]中诚实服务商读取文件块时间：

$$T_h = t_l + q/c \times t_R \tag{4}$$

狡诈服务商读取文件块时间：

$$T_t = t_l + q/c' \times t_R \tag{5}$$

时间差：

$$\Delta t_R = \left(\frac{q}{c'} - \frac{q}{c}\right) \times t_R \tag{6}$$

其中，q 表示请求读取的文件块数，c 表示诚实服务商存储文件使用的磁盘数目，c' 表示狡诈服务商存储文件使用的磁盘数目，t_l 是网络延迟间，t_R 是磁盘寻道时间与读取文件块的总时间，$t_R = t_s + t_r$。根据文献[3]前期研究，云服务商按每块 64 kB 来存储文件效率最高，t_R 平均值约为 6 ms, t_s 平均值约为 3.4 ms，即 $t_R = 1.76 \times t_s$, DRST 和 RAFT 方法的差别如下：

$$\Delta t_D - \Delta t_R = \left\{q \times \left[1 - \frac{t_R}{t_s} \times \left(\frac{1}{c'} - \frac{1}{c}\right)\right] - 1\right\} \times t_s \tag{7}$$

一般情况下云服务商将文件存储在至少两个磁盘上，所以 $c' \geq 2$。

$$\Delta t_D - \Delta t_R \geq \left\{q \times \left[1 - 1.76 \times \left(\frac{1}{2} - \frac{1}{c}\right)\right] - 1\right\} \times t_s \tag{8}$$

用户对服务商的要求一般是将文件存储在至少 3 个磁盘上，即 $c \geq 3$，假设服务器端资源是无限的，即 c 可以取到无穷大，所以式(8)的下界是：$(0.12 \times q - 1) \times t_s$，上界是 $(0.707 \times q - 1) \times t_s$。

当 $n \geq 9$，即文件大小至少是 576 kB 时，$\Delta t_D - \Delta t_R > 0$，此时 DRST 方案对诚实服务商和狡诈服务商的区分度比 RAFT 好。用户在云端存储的文件一般都比较大，大于 576 kB，所以我们的方案在大部分情况下性能优于 RAFT 方案。

为了验证，我们取文献[3]的参数 —— 文件大小：2 GB，块大小：64 kB，服务器承诺使用 4 个磁盘，即 $c = 4$，将一个 2 GB 的文件按照每个文件块 64 kB 均衡存储在 4 个磁盘上，而实际上服务商将文件存在 3 个磁盘上，即 $c' = 3$。t_s 约为 3.4 ms，按照 DRST 方案每步请求一个块，如请求一个磁盘上顺序存储的 51 个块，即 $q = 51$，则

$$\Delta t_D = (q-1) \times t_s = 50 \times 3.4 = 170 \text{ ms} \tag{9}$$

$$\Delta t_R = \left(\frac{q}{c'} - \frac{q}{c}\right) \times t_R = 51 \times \left(\frac{1}{3} - \frac{1}{4}\right) \times 6 = 25.5 \text{ ms} \tag{10}$$

$\Delta t_D \approx 6.7 \Delta t_R$ 说明使用同样的实例参数，DRST 方法理论上比 RAFT 方法更易区分诚实服务商与狡诈服务商，性能更佳。

3.3 DRST 方法性能的测试与分析

文献[1]对 RAFT 方法进行了测试，并在相同网络条件下对诚实服务商和狡诈服务商具体的延迟时间进行了实验，结果表明网络延迟时间分布稳定，因此本文测试实验不关心网络延迟。本小节对 DRST 方法进行测试，与文献[1]中 RAFT 方法的测试结果进行对比。根据 3.2 式(3)及式(6)，要区分诚实服务商与狡诈服务商，DRST 方法与 t_s 即磁盘寻道时间有关，RAFT 方法与 t_R，即磁盘寻道时间与读取文件块的总时间有关，$t_R = t_s + t_r$。本测试旨在对比两个方案的性能，不需要对两个方案相同的网络延迟时间进行测试与比较，只需进行 DRST 的核心步骤即发送请求、读取文件块及记录响应时间，最后与文献[1]中 RAFT 方法的测试时间进行比较。

3.3.1 DRST 测试场景与测试方案

(1)测试场景：　由于不关心网络延迟，我们在本地搭建了测试系统，使用 HP Compad dx7408 MT DT PC 主机，具有 ms-7352 主板，双核 Intel(R) Core (TM)2 Duo CPU E8400 @3.00 GHz, 2 GB 内存，其上挂载 4 个空的 Seagate Barracuda 7200.10.250 G 硬盘，平均寻道时间为 3.4 ms，安装 Ubuntu-14.04 系统，C 语言编写测试程序。

(2)测试方案：　所服务商承诺，在 4 个磁盘上存储一个大小为 2 GB 的文件，根据 DRST 步骤 3 的算法，若服务商是诚实的，将如图 2 所示存储文件，若服务商是狡诈的，就可能使用较少数量的磁盘，如使用 3 个磁盘，如图 3，图中只画出部分块，每块 64 kB。可以看到图 3 每个磁盘上存储的文件块集合与约定的形式有很大差异。该情况下，根据 DRST 方法不难检测出服务商的犯规行为。用户端向服务商提出读取文件块请求，发出多步请求，如依次请求第 1 号，第 5 号，第 9 号，第 13 号…，发出 51 步请求。若服务商是诚实的，按照规定的策略存储文件，那么第 1 号，第 5 号，第 9 号，第 13 号…刚好位于同一磁盘，服务器只需顺序访问 1, 5, 9, 13,… 即可，只需要一次寻道时间。若服务商是狡诈的，本应在一个磁盘上顺序存储的文件块就会分散存储到其他磁盘上，出现一定的随机性，如图 3，此时要读取 1, 5, 9, 13,…，就需要多次寻道时间，使总的响应时间变长。因此，我们可以记录读取文件块的响应时间，若时间在一定的范围内，则认为服务商是诚实的，否则，认为服务商有犯规行为。

3.3.2 DRST 测试数据

按照上述测试方案及参数，

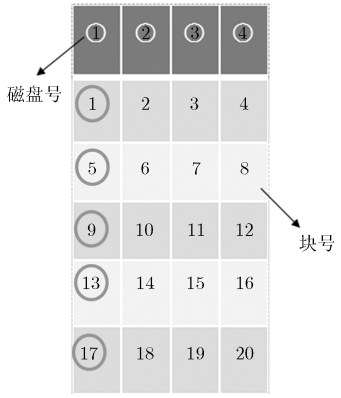

图 2 诚实服务商存储文件形式(4 个磁盘)

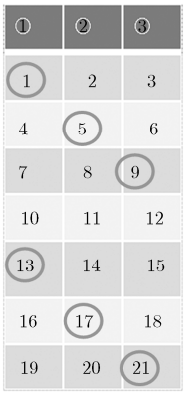

图 3 狡诈服务商存储文件形式(3 个磁盘)

我们进行了 200 次实验得到 200 组数据。本应顺序存放在 1 个磁盘上的文件块被狡诈服务商分散存储到了 3 个磁盘上，用户发出 51 步请求，记录每一步读取文件块的时间，$T1$ 表示随机读取 51 块的总时间。诚实服务商按照承诺将文件分配到 4 个磁盘，我们所请求的文件块刚好存储在 1 个磁盘上，用户同样发出 51 步请求，$T2$ 表示顺序读取 51 块的总时间。将 200 组 $T1$ 和 $T2$ 进行统计，表 1 展示出部分数据。

表 1 51 步狡诈服务商和诚实服务商读取时间

实验序号	$T1$(ms)	$T2$(ms)	实验序号	$T1$(ms)	$T2$(ms)
1	199.522	73.675	101	214.777	74.771
2	199.736	73.325	102	206.179	73.52
3	224.738	74.55	103	207.367	74.603
4	214.021	74.56	104	214.175	74.761
5	206.996	74.503	105	256.604	74.62
6	230.615	74.801	106	214.595	74.751
7	247.911	74.328	107	224.999	74.698
8	213.72	74.847	108	191.404	74.729
9	216.638	74.935	109	213.199	74.785
10	256.736	74.668	110	181.924	83.124

3.3.3 DRST 测试结果分析 根据 200 次实验统计的 200 组数据，我们求出狡诈服务商的平均响应时间 $\overline{T1}$，最短响应时间 $T1_{\min}$，及诚实服务商的平均响应时间 $\overline{T2}$，最长响应时间 $T2_{\max}$。我们将通过这 4 个参数与 RAFT 方法[3]进行对比。

(1)平均情况： 根据统计数据得到 $T1$ 和 $T2$ 分布图，如图 4 和图 5。将 $T1$ 和 $T2$ 画于图 6，每次实验两者均有明显差异，说明我们利用顺序读取和随机读取的寻道时间差来区分诚实与狡诈服务商是可行的且有效的。

根据统计计算得 $\overline{T1}$=226.193 ms, $\overline{T2}$=75.029 ms, $\Delta\overline{T}=\overline{T1}-\overline{T2}=223.934-74.902=149.032$ ms 。

$\Delta\overline{T}$ (149 ms)与 3.2 节理论预测的 DRST 方法用 51 步达到 170 ms 的时间差相差不多，即实际测试和理论预测在一定的范围内是吻合的，说明我们的理论是可信的，测试是可靠的，使得 DRST 方法优于 RAFT 方法的结论更有说服力。

(2)完全区分情况： 根据统计数据得到 $T1$ 和 $T2$ 概率分布图如图 7 和图 8。将 $T1$ 和 $T2$ 画于图 9, $T1$ 和 $T2$ 分布的范围相距较远，更有力地说明我们利用顺序读取和随机读取的寻道时间差来区分诚实与狡诈服务商是有效的。

根据统计计算得 $T1_{\min}=181.924$ ms，$T2_{\max}$= 83.266 ms, $\Delta\overline{T}_{\max}=T1_{\min}-T2_{\max}$=181.924 − 83.266 98.658 ms。

本文的 DRST 方法只需要 51 步就能达到 99 ms 的时间差来完全区分诚实服务商与狡诈服务商，而文献[3]中 RAFT 方法需使用 100 步才能达到 107 ms 的时间差，如图 10。

从以上两方面的数据分析结果，可以得出：

(1)平均情况下，DRST 方法用 50 步达到约 149 ms 的时间差，与理论预测的 170 ms 相差不多，即实际测试和理论预测在一定的误差范围内是匹配

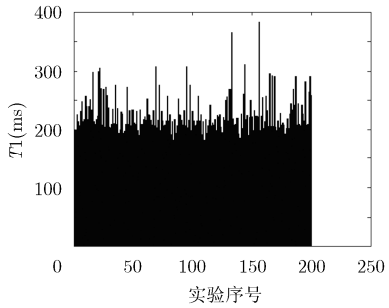

图 4 51 步随机读取时间 $T1$(ms)分布图

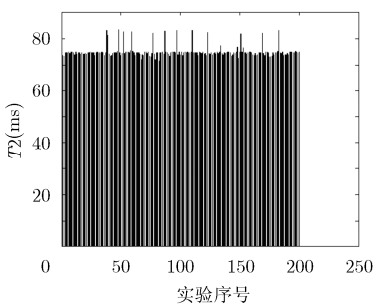

图 5 51 步顺序读取时间 $T2$(ms)分布图

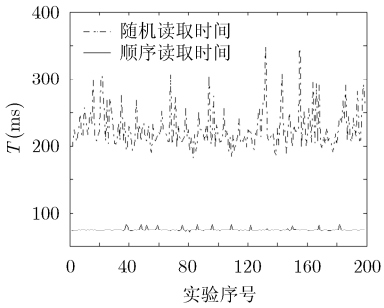

图 6 51 步随机读取时间 $T1$(ms)和顺序读取时间 $T2$(ms)分布图

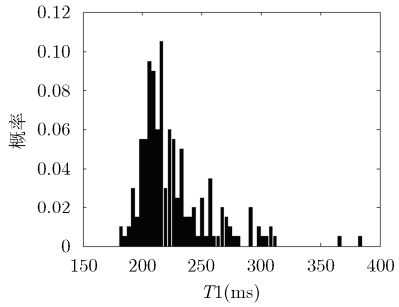

图 7　51 步随机读取时间 $T1$ 概率分布图

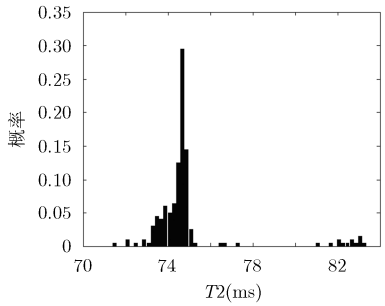

图 8　51 步顺序读取时间 $T2$ 概率分布图

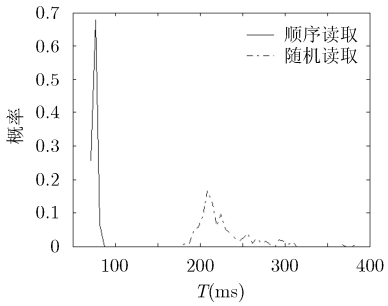

图 9　51 步随机读取时间 $T1$(ms)和顺序读取时间 $T2$(ms)概率分布图

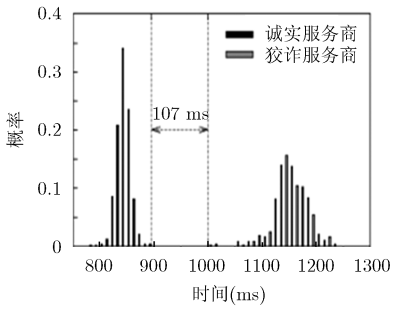

图 10　100 步完全区分诚实服务商与狡诈服务商概率分布图

的，说明本文的理论是可信的，测试是可靠的，进一步说明 DRST 方法优于 RAFT 方法。

(2)完全区分诚实与狡诈服务商的情况下，DRST 方法使用 51 步就可达到约 99 ms 的时间差。相同情况下 RAFT 方法使用 100 步才能达到 107 ms 的时间差。本文方法使用的步数约是 RAFT 方法的一半，效率提高了一倍。

(3)本文采用由用户端提出请求，每步都向服务器说明要读取的文件块序号，服务器并不知道也无法预测用户端下一步要读取哪一块，无法预取来迷惑用户，可防止服务器将预取的文件块返回，DRST 方法更安全。

(4)RAFT 方法中，不论是诚实服务商还是狡诈服务商，进行远程验证时服务器都需要从多个磁盘读取文件，开销大；本文的 DRST 方法中若是诚实服务商，只需要从一个磁盘读取文件，更节省开销。

(5)DRST 方法最重要的价值在于，以测试数据为基准可以检测云存储数据存储方式，即判断服务商是否是按约定将文件均衡存储在多个磁盘的。例如，用户端发出 51 步请求，记录响应时间并验证文件块的正确性，多次测试求平均值，减去实时网络传输时延，若时间落在 71~83 ms 之间则认为是遵守承诺的，若响应时间更长说明服务商有犯规行为，存储文件时使用了较少的磁盘。因此，可事先进行多次实验求平均值，将该平均值作为后续用户进行判断的依据，方便更多的用户方便、轻量级地辨别云服务提供商是否作弊。

综上所述，本文的 DRST 方法效率更高，更安全，能够实现快速、轻量级地检测出犯规的服务器行为的要求，从而判断服务商是否是按照约定将文件均衡存储在多个磁盘，具有很高的应用价值。

4　结束语

云计算作为一种新型的计算模式，在科学计算和商业领域发挥着重要作用，受到当前学术界和企业界的广泛关注。用户对云文件的依赖性日益增长，越来越多的数据存储于数据中心中，但云服务宕机事件越来越多。防止数据损坏和丢失的关键是云存储实现文件容错，云存储供应商通常会按照要求的容错水平收费。然而，云存储服务商可能无法提供承诺的容错水平，用户可能蒙受数据丢失和经济损失。

现有云存储环境下数据容错存储方式的检验方法可能存在服务器预读取的欺骗攻击及效率较低实用性差的缺点。针对该问题，本文在研究已有检验方法的基础上，利用磁盘顺序存取和随机存取的差异性设计了一种远程数据容错存储方式检验方法——DRST，该方法本质是文件块被分散地存储在不同的磁盘上，读取一个磁盘上顺序存储的文件块比读取不同磁盘上的随机块所需的响应时间短。最后

对所提方法进行了严格的理论证明和深入的性能分析，结果表明，本文方法能够快速、轻量级地检测出犯规的服务器行为，从而判断服务商是否是按照约定将文件均衡存储在多个磁盘，具有很高的应用价值。

参 考 文 献

[1] Chinese Institute of Electronics. Future oriented cloud service providers[C]. The Seventh Annual China Cloud Computing Conference, Beijing, China, 2015: 1080–1092.

[2] BARACALDO N, ANDROULAKI E, GLIDER J, *et al*. Reconciling end-to-end confidentiality and data reduction in cloud storage[C]. Proceedings of the 6th ACM Workshop on Cloud Computing Security, Scottsdale, AZ, USA, 2014: 4003–4108.

[3] BOWERS K, DIJK M, JUELS A, *et al*. How to tell if your cloud files are vulnerable to drive crashes[C]. Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS), Chicago, IL, USA, 2011: 2780–2814.

[4] LORENA G and ORFILA A. An efficient confidentiality-preserving proof of ownership for deduplication[J]. *Journal of Network and Computer Applications*, 2015, 50: 49–59.

[5] LI M, QIN C, and LEE P. CDStore: toward reliable, secure, and cost-efficient cloud storage via convergent dispersal[C]. Proceedings of the 2015 USENIX Conference on Usenix Annual Technical Conference, Santa Clara, CA, USA, 2015: 3508–3520.

[6] JUELS A and KALISKI B. PORs – proofs of retrievability for large files[C]. Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS), Alexandia, USA, 2007: 584–597.

[7] SHACHAM H and WATERS B. Compact proofs of retrievability[C]. Asiacrypt 2008, Springer-Verlag, Josef Pieprzyk, 2008: 90–107.

[8] ATENIESE G, BURNS R, CURTMOLA R, *et al*. Provable data possession at untrusted stores[C]. Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS), Alexandia, USA, 2007: 598–609.

[9] DODIS Y, VADHAN S, and WICHS D. Proofs of retrievability via hardness amplification[C]. Theory of Cryptography Conference (TCC), San Francisco, USA, 2009: 235–248.

[10] CURTMOLA R, KHAN O, BURNS R, *et al*. MR.PDP: Multiple-replica provable data possession[C]. Proceedings of 28th IEEE International Conference on Distributed Computing Systems (ICDCS), Beijing, China, 2008: 767–779.

[11] WANG Z, SUN K, JING J, *et al*. Disk storage isolation and verication in cloud[C]. Proceedings of the Globecom, Anaheim, USA, 2012: 898–910.

[12] WANG Z, SUN K, JING J, *et al*. Verification of data redundancy in cloud storage[C]. Proceedings of the International Workshop on Security in Cloud Computing, Hangzhou, China, 2013: 457–468.

纪 倩： 女，1989 年生，博士生，研究方向为云存储安全、云存储文件去重删除.

杨 超： 男，1979 年生，博士，副教授，主要研究领域为大数据与云计算的安全、移动智能计算的安全.

赵文红： 女，1985 年生，硕士，讲师，主要研究领域为无线网络安全、密码学、协议的形式化分析与设计.

张俊伟： 男，1981 年生，博士，副教授，主要研究领域为无线网络安全、密码学、协议的形式化分析与设计.