P-Promise:一个基于承诺的P2P公平资源共享协议*)

胡和平　黄保华　卢正鼎　李瑞轩
(华中科技大学计算机科学与技术学院　武汉 430074)

摘　要　P-Promise借鉴人类社会个体间相互承诺并通过兑现承诺来实现公平交往的机制,通过定义承诺证书和一套协议原语,在P2P实体间建立公平的资源共享环以实现P2P公平资源共享。公平的资源共享环的建立过程就是对等实体承诺并不断兑现承诺的过程,不需要可信第三方、认证的身份、货币支付、对称存储关系等条件。承诺证书能够描述不同的资源,所以P-Promise可用于各种资源的共享。P-Promise不但可以用于各种普通计算机,而且适用于资源和计算能力都十分有限的移动设备。分析和实验证明,P-Promise具有良好的稳定性和抗攻击能力。

关键词　P2P,承诺,公平,资源共享

P-Promise: A Promise Based P2P Fair Resource Sharing Protocol

HU He-Ping　HUANG Bao-Hua　LU Zhen-Ding　LI Rui-Xuan
(College of Computer Science & Technology, Huazhong University of Science & Technology, Wuhan 430074)

Abstract　P-Promise gets inspiration from society that individuals exchange fairly by promising and acting on it. P-Promise defines promise certificate and a suit of protocol primitive. By using promise certificate and protocol primitive, fair resource sharing ring can be built, and the goal of fair resource sharing can be achieved. The process of building fair resource sharing ring is a process of promising and carrying it out, so P-Promise enforces fairness in P2P resource sharing without requiring trusted third parties, certified identities, monetary payment, and symmetric storage relationships. Promise certificate can describe various kinds of resources, so P-Promise can be applied to a wide range of resource sharing. P-Promise can be used not only with general computer, but also with mobile devices that have limited resource and computing ability. Analyses and experiment show that P-Promise is steady and can withstand attack.

Keywords　P2P, Promise, Fairness, Resource sharing

1　引言

P2P(Peer-to-Peer)可以简单地定义为通过直接交换来共享计算机资源和服务[1]。P2P中各节点平等地交互,这有利于增强可靠性和扩展性,有利于资源聚合并减少费用开销,在存储资源、计算周期、内容共享等方面得到广泛应用[2]。P2P通常假定参与者都愿意和别人分享自己的资源,但一项针对Gnutella的研究表明,有70%的用户从不提供共享文件,而数量仅占1%的机器要满足服务总量50%的访问请求[3]。这种现象会导致P2P系统性能的下降并强化系统的弱点,因此,在P2P环境中建立起公平的资源共享机制,奖励资源提供者,保证参与者付出和获得的正比关系是十分必要的。

有关P2P中公平的研究成果一般都要求有可信第三方、对称的存储关系、货币支付或认证的身份等。Samsara是一个可以避免上述限制的P2P公平存储方案[4],但该方案对非存储资源缺乏处理能力,另外该方案也不能适应PDA等资源有限的移动设备的要求。

P-Promise借鉴了人类社会个体间相互承诺并通过兑现承诺来实现公平交往的机制,通过建立公平的资源共享环以实现公平资源共享,环的建立过程就是对等实体承诺并不断兑现承诺的过程。P-Promise不需要可信第三方,也不需要称存储关系和货币支付以及认证的身份,并克服了Samsara的不足。

2　P-Promise协议

P-Promise协议包括承诺证书和协议原语。首先给出承诺证书和协议原语的定义,然后介绍协议工作原理。

2.1　承诺证书

定义1　承诺证书是资源消耗者颁发给资源提供者的证明证书,可以表示为:

$C_{ij}=\langle i,j,R\rangle$

i是证书发布者,即资源使用者;j是证书接收者,即资源的提供者;R是i消耗的j的资源的定量描述,具体形式和语法由应用确定,因此可以描述各种资源。证书由发布者生成并经过签名,别人无法伪造。承诺证书是人类社会书面或口头承诺在计算机世界的表示。

2.2　协议原语

定义2　承诺证书生成原语是资源使用者向资源提供者颁发承诺证书的过程,表示为:

$C_{ij}=promise(i,j,R)$

C_{ij}是生成的承诺证书;$promise$表示原语名称;i是承诺证书发布者,即资源使用者;j是承诺证书接收者,即资源提

*)基金项目:国家自然科学基金(60403027)资助。胡和平　教授,主要研究方向为:软件工程、智能决策系统、信息安全等;黄保华　博士研究生,主要研究方向为:P2P安全与应用等;卢正鼎　教授,博导,主要研究方向为:分布式计算、软件集成环境、数据库、信息安全等;李瑞轩　博士,副教授,主要研究方向为:分布式异构系统集成与安全,Web数据管理,语义网与本体论,对等计算,边缘计算等。

供者；R 是资源的定量描述。该原语由 i 发起并在 i 上执行完成，其它 P2P 参与者无法约束，因此任何参与者都可以执行该原语。

定义 3 承诺转移原语是资源提供者要求资源消耗者将其对自己的承诺转移给第三方，并生成新的对第三方的承诺证书的过程，表示如为：

$C_{ik}=shift\text{-}promise(j,i,k,C_{ij})$

C_{ik} 表示承诺转移后生成的 i 对 k 的承诺证书；*shift-promise* 是原语名；j 是转移动作发起节点；i 是执行转移动作的节点；k 是新的资源提供节点。该原语由 j 发起，j 将 C_{ij} 发送给 i 请求并请求 i 将其对 j 的承诺转移给 k。i 要检查 C_{ij} 的合法性，并在执行完成承诺转移后使 C_{ij} 失效。

定义 4 承诺证书有效性检查原语是资源提供者向证书发布者验证证书真实性的过程，表示为：

$check(k,i,C_{ik})$

check 是原语名称；k 是检查发起节点，就是资源提供者；i 是承诺证书颁发者；C_{ik} 是待验证的证书。当 C_{ik} 是转移承诺证书时需要执行该原语，由 k 发起，在 i 上执行验证动作，结果返回 k。

定义 5 资源使用原语是资源消耗者使用资源提供者资源的过程，表示为：

$use(i,j,R,C_{xj})$

use 是原语名称；i 是资源使用节点；j 是资源提供节点；R 是对资源的定量描述；C_{xj} 是 i 为使用 j 上的资源 R 向 j 提供的承诺证书。如果 $x=i$，C_{xj} 不是转移承诺证书，则 j 可以直接使用 C_{xj}，否则 j 必须执行 $check(j,x,C_{xj})$ 向 x 验证 C_{xj} 的有效性。该原语执行涉及 i,j,x 三方。

定义 6 资源删除原语是资源提供者撤消资源使用者资源使用资格的过程，表示为：

$delete(i,j,R)$

delete 是原语名称；i 是执行撤消的节点；j 是被撤消资源使用资格的节点；R 是对资源的定量描述。该原语在 i 上执行完成，并不一定删除实际的资源内容，但结果一定导致 j 不能再使用 i 上的资源 R。

定义 7 资源状态查询原语是资源使用者为确保资源提供者提供的资源可靠而定期或不定期检查的过程，表示为：

$query(i,j,R)$

query 是原语名称；i 是执行查询的节点；j 是被查询的节点；R 表示对资源的定量描述。该原语执行涉及 i,j 两方。

2.3 工作原理

P-Promise 工作原理如图 1 所示。R 所在行表示节点资源使用情况，C 所在行表示节点颁发的承诺证书。1,2,3,…，n 表示 1 到 n 个 节点。不失一般性，设节点 1 需要使用节点 2 上的资源 R，则节点 1 执行 $C_{12}=promis(1,2,R)$ 和 $use(1,2,R,C_{12})$，执行结果如图 1(1)所示。接下来节点 2 要使用节点 3 上的资源，因为 2 已经向 P2P 环境做出了贡献并得到了其它节点的承诺，因此，它执行 $C_{13}=shift\text{-}promise(2,1,3,C_{12})$ 请求节点 1 将承诺转移给节点 3，然后执行 $use(2,3,R,C_{13})$，节点 3 执行 $check(3,1,C_{13})$ 检查以确保 C_{13} 是节点 1 对 3 的承诺，执行结果如图 1(2)所示。以此类推，一直到 n 要使用节点 1 上的资源，这时节点 n 已经有节点 1 给它的承诺证书 C_{1n}，它直接执行 $use(n,1,R,C_{1n})$ 来使用节点 1 上的资源，执行结果如图 1(n)所示。至此，一个公平的资源使用环形成。

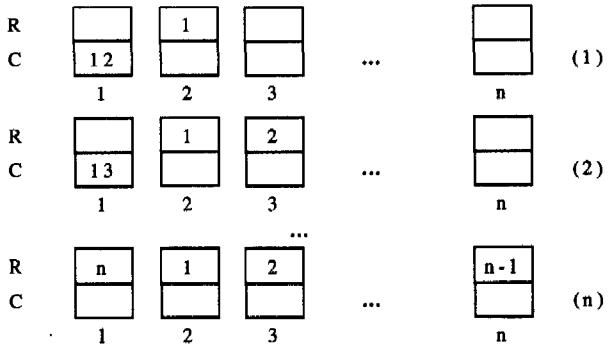

图 1 P-Promise 工作原理

公平资源使用环在形成过程中借助了承诺证书，当环形成后，各节点都用行动实现了自己的承诺，环中各节点的资源使用是公平的，承诺证书消失。

公平资源使用环正常工作过程中，节点 i 通过 $query(i,j,R)$ 来检查节点 j 上资源 R 的状态，如果异常，则表明节点 j 已经不遵守其承诺，公平资源使用环破裂，节点 i 也不能再为其上游节点 $i-1$ 提供资源，因此节点 i 使用 $delete(i,i-1,R)$ 删除节点 $i-1$ 对节点 i 上资源的使用资格并重新在 P2P 环境中寻找合适的资源使用。这将依次导致资源使用环中所有节点执行这一过程，导致资源的重新部署。

设节点 x 失效给系统带来的损失可以表示为：$Cost=\sum_{i=1,i\neq x}^{n} f_i$，为讨论简便，设各节点损失 f_i 相同并用 f 表示，则 x 失效给系统带来的损失为：

$$Cost=(n-1)f \tag{1}$$

上述 P-Promise 实现简单，公平资源使用环形成后各节点不需要保存承诺证书。但环中任一节点出现问题都会导致整个环破裂并重新部署资源，给计算能力、网络资源、设备 IO 等带来巨大压力，性能较差。

把上述 P-Promise 称为基本 P-Promise(Basic)，接下来扩展 P-Promise 以解决性能问题。

3 扩展 P-Promise

当节点 i 检测到 j 失效时，它可以只废除其对下游节点的承诺而继续保持对上游的承诺，也可以将 j 旁路，建立一个不包含 j 节点的新的环。这两种方式都需要各节点保留承诺证书的生成和使用历史。

3.1 保持对上游节点的承诺

定义 8 承诺转移撤消原语是资源提供者要求证书颁发者撤消承诺转移的过程，表示为：

$dis\text{-}shift(j,i,C_{ik})$

dis-shift 是原语名称；j 是撤消转移的发起节点；i 是证书颁发节点；C_{ik} 是 j 请求 i 撤消的已经转移给 k 的承诺证书。该语言原语由 j 发起，由 i 执行实际的撤消操作。执行结果使 C_{ij} 有效，而直接或间接通过 C_{ij} 转移的承诺证书无效。

当 i 检测到 j 不能提供承诺的资源时，i 执行 $dis\text{-}shift(i,1,C_{1j})$，导致 C_{1i} 有效而 C_{1j}，C_{1k}，…，C_{1n} 失效，1 将执行 $delete(1,n,R)$，导致 n 在 1 上的资源使用失败，依次导致 $n,n-1,\cdots,k$ 节点的资源重新调整。i 节点则可执行 $C_{1x}=shift\text{-}promise(i,1,x,C_{1i})$ 和 $use(i,x,R,C_{1x})$ 来使用重新选择的 x 上的资源。

节点 j 失效给系统带来的损失可表示为 $(n-j)f$，节点失

效造成的平均损失表示为 $Cost=\sum_{j=1}^{n} p_j(n-j)f$，$p_j$ 表示节点 j 失效的概率。设 p_j 均等，则

$$Cost=\frac{1}{2}(n-1)f \quad (2)$$

对比式(2)和(1)可以看出增加承诺转移撤消原语可以减少一半损失。

3.2 节点旁路

定义 9 承诺失效通知原语是资源使用者将资源提供者承诺失效的事实通知其它节点的过程，分为两种情况。当失效节点不是 1 时，表示为：

$C_{ki}=notify(i,1,C_{1j})$

执行过程由 i 发起并提交 1 执行，1 再提交 k 执行并将 k 返回的结果传递给 i。

当失效节点是 1 时，$i=n$，表示为：

$C_{2n}=notify(n,n-1,C_{1n})$

执行过程由 n 发起，依次经过 $n-1,n\cdots2$ 节点，结果逆序依次返回给 n。

当 i 检测到 j 失效。如果 $j\neq1$，i 执行 $C_{ki}=notify(i,1,C_{1j})$，然后执行 $use(i,k,R,C_{ki})$ 使用 k 上的资源，将失效节点 j 旁路。如果 $j=1,i=n$ 执行 $C_{2n}=notify(n,n-1,C_{1n})$，然后执行 $use(n,2,R,C_{2n})$ 将 1 旁路。

因为节点失效不引起资源的重新部署，因此给系统带来的损失基本为 0，即：

$$Cost=0 \quad (3)$$

两种扩展方式相比，前者性能比后者差，但前者在执行时涉及的节点数量少，后者涉及节点数量多，并且不一定符合 P2P 资源使用选择的要求。如节点 k 不一定满足节点 i 在信任度方面和性能方面的要求，因此 i 不能直接选择 k 作为其资源提供者，不能进行节点旁路。

4 安全问题

承诺滥用和拒绝服务攻击是 P-Promise 面临的两个最重要的安全问题。承诺滥用会破坏 P-Promise 的资源公平共享目标，而拒绝服务攻击则会带来系统安全问题。P-Promise 和 P2P 环境的信任管理方案一起使用不但可以有效避免 P-Promise 的安全问题，同时还能够完善信任管理方案，使信任管理能够反映和惩罚不良行为。

4.1 承诺滥用

在 P-Promise 中，任何节点都可执行 promise 生成承诺证书，而公平资源使用环的形成需要一定时间，恶意节点可以借用这段时间滥用 promise 原语来获得不需兑现承诺的免费资源使用，破坏了 P-Promise 的公平性。

为防止这种权利被滥用，可将 P-Promise 和 P2P 系统中的信任管理机制结合使用。一方面 P-Promise 可以使用信任管理方案提供的节点可信度来衡量其提供的承诺证书的可靠性，对不满足要求的节点提供的承诺证书拒绝 use 原语。另一方面，P2P 信任管理方案可以对 P-Promise 发现的不遵守承诺的节点进行惩罚，降低其信任度，使信任管理对恶意行为更敏感。

4.2 拒绝服务攻击

对基本 P-Promise 和保持上游节点承诺的扩展，节点的失效将导致资源的重新部署，带来计算、网络和设备 IO 等开销，恶意节点可以利用这一特点来消耗资源，影响或破坏正常服务。实际上，这些恶意节点必然不遵守承诺，P-Promise 结合信任管理方案，可以使信任管理方案对这种恶意节点更敏感，从而将恶意节点排除在资源使用环外，有效防止这种攻击。

5 应用与实验

为检验 P-Promise，它被应用到一个 P2P 备份系统中。该备份系统将各计算机上的空闲存储空间组织起来进行数据的相互备份，本质上是共享存储资源。系统结构如图 2。

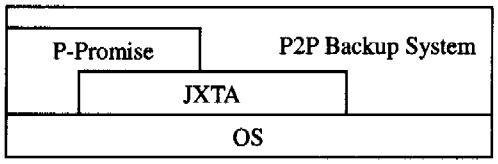

图 2 P-Promise 用于 P2P 备份系统

JXTA 是一套允许各种设备以 P2P 方式进行通信和合作的开放协议，有开放源代码可以使用。系统中采用节点公钥标识节点，公钥肯定是唯一的。节点公/私钥对是节点自己生成的，公钥直接提供给通信的另一方，私钥自己保存，不需要第三方认证。考虑到目前计算机存储容量都比较大，R 定义为〈Disk Space n MB〉。证书格式为〈PK_i，PK_j，R〉，PK_i 是 i 的公钥，PK_j 是 j 的公钥。系统中将局部可信度(节点自己对其它节点的评价)和 P-Promise 配合使用。

实验使用 4 台计算机，在每台计算机上运行 10 个进程模拟 10 个节点，共模拟 40 个节点，其中随机选择 4 个扮演恶意节点，不为其他节点提供服务。实验结果显示，P-Promise 使各节点的磁盘空间贡献和使用相等，实现了存储空间公平共享的目标。基本 P-Promise(basic)和两种扩展方式(dis-shift，notify)I 的 IO 增量随时间关系如图 3。

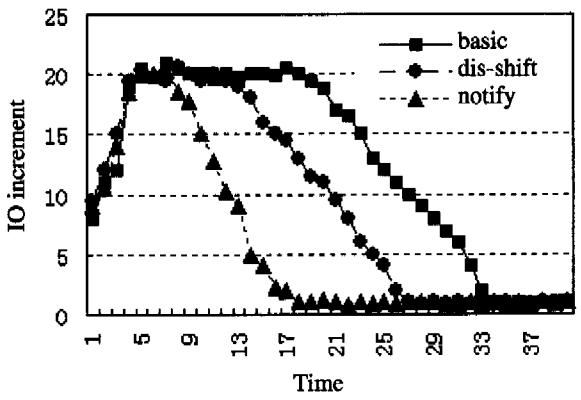

图 3 IO 增量时间关系

从图中可以看出，基本 P-Promise 及其扩展的 IO 增量都能够稳定，说明和信任管理结合使用的 P-Promise 具有抗攻击能力。节点旁路扩展优于保持上游承诺的扩展，而保持上游承诺的扩展又优于基本 P-Promise，这与前面的分析是一致的。

6 相关工作

FARSITE 提出了应该保证节点所消费的资源等于或小于其所贡献的资源的思想，但没有给出具体的方案[5]。CFS 中节点将其它节点可使用的存储资源限制在一个固定的比例下，节点配额与其贡献无关[6]，并通过IP来标识节点，抗攻

(下转第 61 页)

API:GSocket。GSocket在不同的网络系统的顶端实现了一个智能的策略控制模块,使网络应用程序具有很好的适应性和可扩展性能,在统一接口界面下,能够很好地适应网络应用和网络技术的发展。这为我们开发新的网络体系结构提供了一个很好的编程接口。经过比较测试其模型的运行性能,发现其运行性能很接近目前使用的BSD socket的性能。

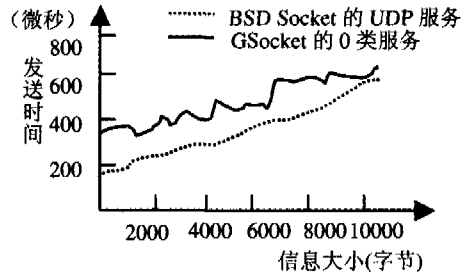

图3 0类服务与UDP的发送时间对比

下一步需要做的工作是根据实际的需要定义更多的网络服务类型,并实现其模型进行性能测试和比较,同时优化设计套接字策略模块,进一步提高GSocket的处理性能。

参考文献

1 Rosen E, Viswanathan A, Callon R. Multi- protocol Label Switching Architecture. RFC3031, Jan. 2001
2 Braden R, Zhang L, Estrin D, et al. Resource ReServation Protocol (RSVP) Version 1 Functional Specification. RFC2205, September 1997
3 Schulzrinne H, Casner S, Frederick R, et al. RTP: a Trans- port Protocol for Real-Time Applications. RFC 1889, 1989
4 De Prycker M. Asynchronous Transfer Mode: Solution for Broadband ISDN, 3rd. Prentice-Hall, Englewood Cli. s, NJ, 1995
5 Braden B, Faber T, Handley M. From Protocol Stack to Protocol Heap - Role-Based Architecture. First Workshop on Hot Topics in Networking, Oct. 2002
6 曾家智,等. 服务元网络体系结构和微通信元系统构架. 电子学报 2004,32(5):745~749
9 Bocking S. Sockets++: a uniform application programming interface for basic level communication services. Communications Magazine, IEEE, 1996, 34(12): 114~123
10 Bocking S. TIP's Performance Quality of Service. IEEE Commun. Mag. 1993, 33(8)
11 Florissi P G S, Yemini Y, Florissi D. QoSockets: a new extension to the sockets API for end-to-end application QoS management. Computer Networks 2001, 35(1): 57~76
12 Hasan Abbasi, et al. A Quality-of-Service Enhanced Socket API in GNU/Linux. http://www. linuxdevices. com/articles/ AT8639420091. html
13 Microsoft. Windows Socket 2 specication. URL: http://msdn. microsoft. com /library/ psdk/winsock/ wsapiref_2qr6. htm

(上接第49页)

击能力差。PAST提出了一个通过智能卡来实施配额的方案,这需要可信第三方和智能卡硬件[7]。Data trading要求节点间交换相同数量的实际数据[8]和文[9]一样要求对称存储关系。SHARP要求在离线状态下建立起节点间公钥的信任关系[10]。文[11]讨论了存储资源和带宽限制机制,但要求有PKI支持。Fileteller[12]使用Micropayments[13]进行文件存储的收费,但Micropayments要求分布式事务,对P2P备份系统并不适用[14]。

Samsara[4]与P-Promise最接近,其基本思想是通过引入存储声明(storage claim)来将不具备对称存储关系的环境转换为具有对称存储关系的环境来实现公平的存储资源共享。P-Promise和Samsara相比,特点和不同主要有:(1)思想和原理不同,P-Promise借鉴人类社会的承诺与诺言兑现机制,而Samsara是通过存储声明将非对称存储关系转换为对称存储关系,因此,其必然要求参与各方在资源能力方面是对称的,从而限制其适用范围;(2)P-Promise的承诺证书可以描述存储资源以外的资源,如处理器周期、信息等,可以由应用定义,而Samsara的存储声明只适用于存储资源;(3)P-Promise可以适用于资源和处理能力都比较有限的移动设备,比如资源丰富的桌面计算机可以代替移动设备进行承诺并兑现,移动设备就可以使用P2P环境中丰富的资源,而Samsara不能。

结论 P-Promise借鉴人类社会个体间相互承诺并通过兑现承诺来实现公平交往的机制,通过定义承诺证书和协议原语在P2P环境中建立公平的资源共享环并实现P2P公平资源共享。承诺证书可以描述各种资源,所以P-Promise可用于不同资源的共享。公平资源共享环的建立过程就是对等实体承诺并不断兑现承诺的过程,因此,P-Promise不需要可信第三方、认证的身份、货币支付和对称存储关系等有违P2P初衷的条件。分析和实验表明,P-Promise和信任管理机制一起使用,两者相互补充,具有良好的稳定性和抗攻击能力。下一步将把P-Promise应用到存储资源之外的资源共享中。

参考文献

1 Barkai D. An Introduction to Peer-to-Peer Computing. Intel Developer Update Magazine, 2000
2 Milojicic D S, Kalogeraki V, Lukose L, et al. Peer-to-Peer Computing: [Technical report, HPL-2002-57R1]. Hewlett-Packard Company, 2002
3 Adar E, Huberman B A. Free riding on Gnutella. First Monday, 2000, 5(10)
4 Cox L P, Nobel B D. Samsara: Honor Among Thieves in Peer-to-Peer Storage. 19th ACM Symposium on Operating Systems Principles, New York, 2003
5 Adya A, Bolosky W J, Castro M, et al. FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. 5th Symposium on Operating Systems Design and Implementation, Boston, MA, 2002
6 Dabek F, Kaashoek M F, Karger D, Morris R, Stoica I. Wide-area cooperative storage with CFS. 18th ACM Symposium on Operating Systems Principles, Canada, 2001
7 Rowstron A, Druschel P. Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. 18th ACM Symposium on Operating Systems Principles, Banff, Canada, 2001
8 Cooper B F, Garcia-Molina H. Peer-to-peer resource trading in a reliable distributed system. First International Workshop on Peer-to-Peer Systems, Cambridge, MA, 2002
9 Lillibridge M, Elnikety S, Birrell A, Burrows M, Isard M. A cooperative Internet backup scheme. USENIX Annual Technical Conference, Texas, 2003
10 Fu Y, Chase J, Chun B, Schwab S, Vahdat A. SHARP: An architecture for secure resource peering. 19th ACM Symposium on Operating Systems Principles, New York, 2003
11 Tsuen-Wan "Johnny" Ngan, Nandi A, Singh A, et al. Designing Incentives-Compatible Peer-to-Peer Systems. 2nd Bertinoro Workshop on Future Directions in Distributed Computing, Bertinoro, Italy, 2004
12 Ioannidis J, Ioannidis S, Keromytis A D, et al. Fileteller: Paying and getting paid for the storage. 6th Annual Conference on Financial Cryptography, Bermuda, 2002
13 Blaze M, Ioannidis J, Keromytis A. Offline micropayments without trusted hardware. 5th Annual Conference on Financial Cryptography, BWI, 2001
14 Cox L P, Murray C D, Noble B D. Pastiche: Making backup cheap and easy. 5th Symposium on Operating Systems Design and Implementation, Boston, MA, 2002