DCR²S:分布式协作冗余复制存储机制*)

周 旭 卢显良 魏青松
(电子科技大学计算机学院 成都610054)

摘 要 文件复制和编码校验是分布式文件容错中常用的方法。结合两者的优点,本文提出了一种分布式协作冗余复制存储机制(DCR²S)。DCR²S通过XOR校验文件实现了分布在不同主机上的多个文件之间的相互协作,使得各个文件不仅可以通过复制冗余复本来提高自身的容错抗毁性能,并且可以通过检验文件协助其他文件提高容错性,既提高了单个文件的容错性能,更大大提高了一组文件的整体容错性能。本文对DCR²S的原理进行了图论表述,给出了概率计算公式,定量地分析了DCR²S的容错性能。通过计算比较,DCR²S的容错性能远高于完全复制。

关键词 分布式,容错,冗余,协作,图论,校验

DCR²S: Distributed Cooperative Redundancy Replication Storage Mechanism

Zhou Xu LU Xian-Liang WEI Qing-Song
(Department of Computer Science of UEST of China, Chengdu 610054)

Abstract File replication and coding are two common method used in fault-tolerance of distributed file systems. Combining the advantage of replication and coding, this paper presents a novel distributed cooperative redundancy replication storage mechanism (DCR²S). By using XOR coding, DCR²S makes a group of files which distributed among different hosts cooperative, so that not only a single file in the group can using XOR files to improve its own availability, but also the total availability of the whole group can be improved greatly. Under the graph theory description of DCR²S, author gives a quantitative analysis to DCR²S' performance. Comparing to complete replication method, DCR²S has much higher fault-tolerance performance.

Keywords Distributed, Fault-tolerant, Redundancy, Cooperative, Graph theory, XOR

1 引言

随着现代网络技术的不断发展以及计算机存储计算能力的不断提高,使得在广域网内集合大量计算机的存储资源,构建大规模的分布式网络文件存储系统成为可能。目前,很多研究机构和公司都提出了各自不同的分布式网络文件系统结构,其中比较著名的有FreeHeaven[1], Farsite[2], OceanStore[3],FTDSS[4]和PAST[5]等。

但是,随着系统规模的不断扩大,加入的机器不断增加,在增加了整个系统资源数量的同时,也增加了文件系统管理的复杂度和不确定性。如何有效地提高分布在众多机器上的文件数据的可靠性和可用性,使得存放在大量相互独立的计算机上的资源得到更加安全高效的利用,已经成为大规模分布式存储系统中重要的研究课题之一。

为了获取文件的高可用性和高容错性,将文件数据进行各种形式的冗余并分布在存储网络中多个节点上是目前使用得最多的方法之一。很多系统都是基于这种思想:PAST和Farsite采用了完全复制(Complete replication)的方式;FTDSS采用了文件分块(segment)加XOR校验的方式;FreeHeaven和OceanStore则采用了文件分块加Erasure Codes编码[6]。

本文提出了一种新颖的分布式协作冗余复制存储机制——DCR²S(Distributed Cooperative Redundancy Replication Storage Mechanism)。DCR²S结合了完全复制和XOR校验的思想,不仅将文件复制成若干个复本(replica,指同一个文件的多个拷贝),同时还与其他一个或多个文件(协作文件)生成XOR校验文件。这些复本文件和校验文件被分布在系统中不同的机器上。这样,即使一个文件所有的原始复本都不可获得,仍然可以使用协作文件和校验文件通过计算恢复文件数据,实现比单纯复制更好的文件容错性能。

本文第2节介绍DCR²S的基本原理;第3节分析DCR²S的容错性能;最后是结论和未来的工作。

2 DCR²S的基本原理

2.1 常见文件数据容错方法

完全复制、FTDSS和Erasure Codes编码这几种文件数据冗余容错方法原理各不相同,其空间复杂度、编码效率、文件容错性、读写效率也是各有千秋[7]。完全复制思想简单,将文件的多个复本分布到不同的机器上实现冗余容错,复本管理的空间复杂度小,且不涉及编码运算,效率高,容错性能较好,但是要想提高文件的容错性能,只能通过增加文件的冗余度来实现,占用存储资源较多;FTDSS将文件分片,并在这些分片之间相互进行异或运算,再将原始数据片和校验片分布到不同的机器上实现数据的容错,其使用的XOR编码效率高,容错性能较好,但是最后生成校验分片过多,增加了文件管理的空间复杂度,且要提高文件的容错性能,只能更加细分文件分块,这会导致占用的存储资源和空间复杂度快速上升;Erasure Codes也是采取的文件分块校验、分片存储的方法,与FTDSS不同的是它的校验分块是由原始数据分块经过Erasure Codes编码后生成的,这是一种十分复杂的编码方

*)本文受国家95重点攻关项目支持。周 旭 博士研究生,主要研究方向:计算机网络、网络存储、分布式操作系统等。卢显良 教授,博士生导师,主要研究方向:计算机网络、操作系统。魏青松 博士研究生,主要研究方向:计算机网络、网络存储。

式，在和 FTDSS 同样的分片数量下可以实现更小的空间复杂度和更好的容错性能，但是由于编码效率较低，一般只用于只读文件的容错。

研究以上几种文件冗余容错方案，我们可以发现它们都是针对单个文件进行独立的冗余容错，没有考虑在文件之间进行协作。同时我们知道，一个系统中的某些文件之间往往存在某种内在的相互依赖和关联关系，特别是在分布式文件系统中，某个应用可能需要同时读取分布在不同的机器上的几个文件，如果这些文件中的一个或多个受损或丢失，应用往往不能运行或者功能受到影响。所以，对于分布式文件系统来说，我们不仅需要考虑单个文件的容错性能，同时也应该考虑多个分散而又有关联的文件的整体容错性能。

2.2 DCR²S 的基本原理和图论表示

分布式协作冗余复制存储机制 DCR²S 正是基于以上思想的一种新的分布式容错机制，它结合了完全复制和 XOR 校验的方法，将分布在不同机器上的多个文件通过相互之间的校验文件联系起来，形成一个整体，各个文件不仅可以通过复制冗余复本来提高自身的容错性能，并且可以通过校验文件协助其他文件提高容错性。通过多个主机上的文件的相互协作，不仅可以提高单个文件的容错性能，更可以使得这一组文件的整体容错性能得到大大提高。

如图1.a 所示，设文件 A 和 B 是系统中的原始文件(相对于校验文件而言)，其中 A 文件有 m 份复本($A_1,\cdots,A_m$)，B 文件有 n 份复本($B_1,\cdots,B_n$)，A XOR B 表示 A 和 B 通过异或运算生成校验文件，这些文件都分布在不同的主机上。由异或运算的性质可以知道，当存放 A 文件复本的 m 台主机都出现故障或不在线，只要校验文件和任意一个 B 文件的复本在线，仍然可以使用这两者进行异或运算将 A 恢复。同理，B 文件损失后也可以借由校验文件和 A 文件复本进行恢复。

为了表示和计算的方便，我们将所有的 A 文件复本记为一个点 A^m，B 的复本记为点 B^n，将文件 A 和 B 异或运算后生成的校验文件 A XOR B 表示为连接 A^m 和 B^n 两点的一条边 a。这样，借用图论中“图”的概念，图1.a 中几个文件的关系可以简化地表示为图1.b。

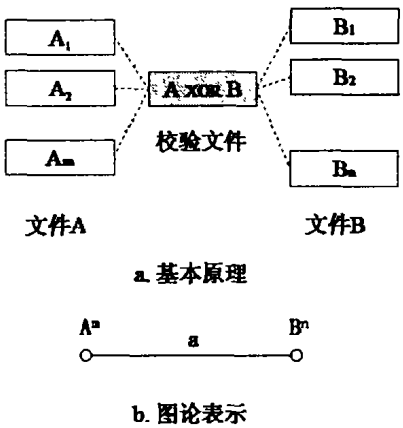

图1 DCR²S 原理和图论表示

如图2.a 所示，我们将一组用校验文件联系起来的分散文件称为一个 DCR²S 单元(DCR²S unit)。在系统的任意时刻，单元中的某些校验文件可能由于所在主机的原因而导致不能获得，这表现在该单元对应的图中就是一部分边的缺失。所以，在任意时刻，单元对应的图都可能只是原来完整时的一个子图。图2.b 就是在某个时刻，由于图2.a 中两条边丢失而形成的子图。

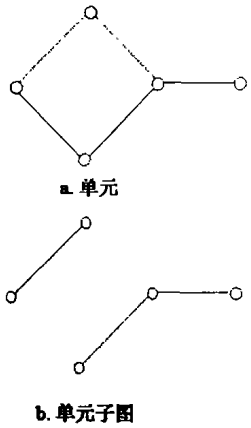

图2 DCR²中的单元

为了叙述的方便，下面给出几个定义。在一个单元中，每个点都有两种状态：若点 A^m 对应的文件 A 的所有 m 个复本都不在线时，称点 A^m 不在线，反之称点 A^m 在线。若一个点 A^m 不在线但是可以通过 XOR 编码重新生成，称点 A^m 是可恢复的。若一个点 A^m 在线或是可恢复的，称点 A^m 可得。如果一个单元中所有点都是可得的，称该单元是整体可得的。

有了这些定义，就可以使用图论的表述方式来准确地描述 DCR²S 中一个单元单点和整体的可恢复条件：

(1)在一个单元中，点 A^m 是可恢复的充要条件是当前单元的图中至少存在另一个在线的点 B^n 并且存在一条连接 A^m 和 B^n 的道路。

(2)一个单元是整体可得的充要条件是当前该单元的任意连通子图中都至少有一个点是在线的。

3 DCR²S 的容错性能分析

3.1 环形和链形 DCR²S 结构

对于多个文件，可以有链形、星形、环形、树形和更加复杂的网形等多种方式将它们相互连接起来，形成一个 DCR²S 单元(如图3所示)。这些连接方式的空间复杂度和占用的存储资源各不相同，容错性能也有差别。下面我们重点讨论链形(chain)和环形(ring)结构，并定量地分析一下这两种结构的容错性能。

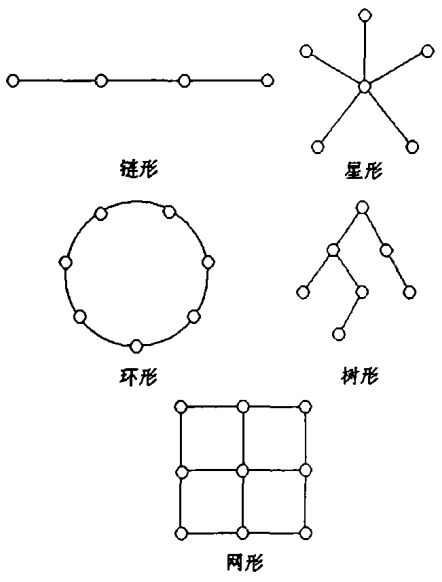

图3 几种不同的 DCR²S 单元结构

为了简化模型，假设系统中每台主机出现故障的概率均

为 p，每个文件有 k 份复本(即文件在系统中共有 k 份拷贝)，每个校验文件在系统中保存的份数只有一份，所有这些文件都分布在不同的主机上。

设单元中每个点不在线的概率为 P，每条边不在线的概率为 Q。则根据定义，有 $P=p^k$，$Q=p$。

设 $Chain(n)$ 表示有 n 个点 $(a_1\ a_2\cdots a_n)$ 的链形单元结构，记 $Chain(n)$ 中单点 a_i 可得的概率为 $c_{n,i}$，记 $Chain(n)$ 整体可得的概率为 C_n。则根据概率论原理，可以求出：

$c_{1,1}=1-P$，

$c_{2,1}=c_{2,2}=(1-P)+P\cdot(1-Q)\cdot c_{1,1}$

当 $n\geqslant 3$ 时，用递推法(推导过程从略)可以得出：

$c_{n,1}=c_{n,n}=(1-P)+P\cdot(1-Q)\cdot c_{n-1,1}$

$c_{n,i}=(1-P)+P\cdot((1-Q)\cdot c_{i-1,1}+(1-Q)\cdot c_{n-i,1}-(1-Q)^2\cdot c_{i-1,1}\cdot c_{n-i,1})$，$i=2,\cdots,n-1$

同理可以计算出 $Chain(n)$ 的整体可得概率：

$C_1=(1-P)$，

$C_2=(1-P)^2+2\cdot P\cdot(1-P)\cdot(1-Q)$

…

$C_n=(1+P-2PQ)\cdot C_{n-1}-P\cdot(1-Q)^2\cdot C_{n-2}$，$n\geqslant 3$

设 $Ring(n)$ 表示有 n 个点 $(a_1\ a_2\cdots a_n)$ 的环形单元结构。很明显，$Ring(n)$ 中每个点的单点可得概率都一样，记为 r_n，另用 R_n 表示 $Ring(n)$ 的整体可得概率。则

$$r_n=(1-P)+\sum_{m=1}^{n-1}P\cdot(1-P)\cdot P^{n-2}\cdot((1-Q)^m+(1-Q)^{n-m}-(1-Q)^n)+\sum_{j=2}^{n-1}\sum_{m=1}^{n-j}P\cdot(1-P)^2\cdot P^{n-j-1}\cdot((1-Q)^m+(1-Q)^{n-m-j+1}-(1-Q)^{n-j+1}),\ n\geqslant 3$$

同样使用递推法，可以得出

$R_3=1-P^3-3P^2(1-P)(Q^3+3Q^2(1-Q))-3P(1-P)^2Q^2$

…

$$R_n=2\frac{1-P\cdot Q}{1-P}C_n+\frac{1+P+P\cdot Q^2+P^2\cdot Q^2-4P\cdot Q}{P-1}C_{n-1}+P^{n-1}\cdot(1-Q)^{n-1}\cdot(3P\cdot Q-P-Q-1)+\sum_{m=1}^{n-3}(n-m-2)\cdot(1-P)\cdot(1-Q)^{n-m-1}\cdot P^{n-m-1}\cdot Q^2\cdot C_m,\ n\geqslant 4$$

从直观上分析，环形单元结构的单点可得概率和整体可得概率都应该比链形结构大，下面通过计算来验证这一点。

假设每台主机出现故障的概率为10%，每个文件只有一份，即 $p=0.1$，$k=1$，可得 $P=0.1$，$Q=0.1$。

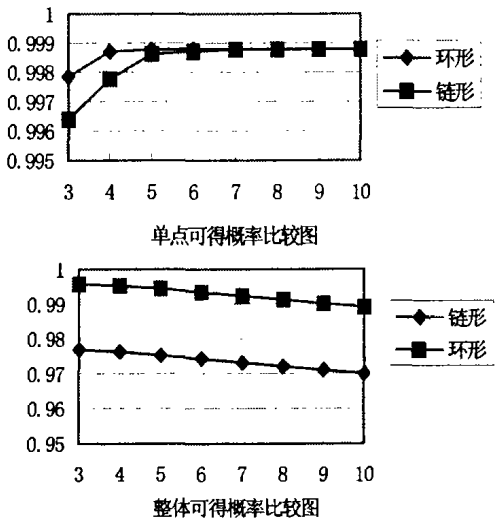

图4 环形和链形结构的容错性能对比

对于 $n=3,\cdots,10$，根据前面的公式可以计算出两种结构的单点和整体可得概率，如图4所示(由于链形结构中各点的可得概率不同，取其最大值)。

从图4中可以看出，DCR²S 的单点可得概率随着 n 的增加而增加，整体可得概率随 n 的增加而缓慢减小，且环形单元结构的单点和整体可得概率一直大于链形结构。当 $n\geqslant 7$ 时，环形结构的单点可得概率的增长变得十分微小，与链形结构单点可得概率之间的差距也越来越小，逐渐趋于一致。这说明环形和链形单元中单点的可得概率主要受图中距离它比较近的点的影响，当单元中的点数增加到一定数量时，继续增加下去并不能有效地提高单点的可得概率。

3.2 完全复制和环形结构 DCR²S 的容错性能比较

DCR²S 是从完全复制的思想发展而来，下面比较一下在占用相同存储空间的条件下，完全复制和环形结构 DCR²S 容错性能的差别。

以下仍然设 $p=0.1$，并假定单元中每个文件的大小都相同，设为一个存储单位。则 $k=1$ 时，点数为 n 的环形 DCR²S 单元将占用 $2n$ 个存储单位。

同样占用 $2n$ 个存储单位，使用完全复制的方法，我们为这 n 个原始文件每个再复制一个复本。可以求出这时每个文件的单点可得概率为 $1-p^2$，而 n 个文件的整体可得概率为 $1-(p^2)^n$。

图5是 n 取不同值时，环形 DCR²S 结构和完全复制的容错性能比较图。从图中可以看出，对于所有的 n 来说，在同样占用 $2n$ 个存储单位时，环形 DCR²S 结构的单点可得概率和整体可得概率都大于完全复制。另外，由于环形 DCR²S 结构整体可得概率下降的趋势大大缓于完全复制，随着 n 的增加，两者之间的差距越来越大，环形 DCR²S 的优势愈加突出。

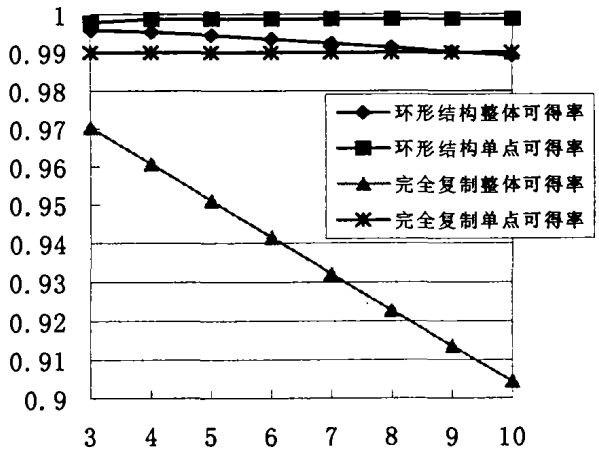

图5 环形结构和完全复制冗错性能比较

以 $n=5$ 为例，如图6所示，设有5个分布在不同主机上的文件，它们的单点可得概率为0.9，整体可得概率等于 $0.9^5=0.6561$。若使用完全复制的方法，为每个文件增加一个复本，则文件的单点可得概率为0.99，整体可得概率为0.9509900499；若使用环形 DCR²S，$k=1$ 时文件的单点可得概率为0.9987835986，整体可得概率为0.9945327825。所以，同样增加5个存储单位建立校验文件形成一个环形单元，可以将文件的单点容错性能提高到不容错时的 $\frac{1-0.9}{1-0.9987835986}\approx 82.21$ 倍，将整体容错性能提高到不容错时的

(下转第213页)

的路径，如果这些路径可以被使用，那么反向页表的开销则可以避免，如图4所示。

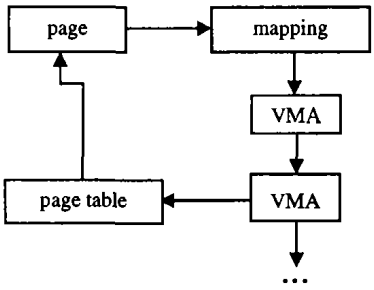

图4 文件映射页面的一种反向映射路径

page 结构中有一项指向 address_space 结构的域 mapping，该域描述了备份这个页面的文件。address_space 结构包括备份文件的 inode、文件页面的数据结构和两个 vm_area_struct(VMA)的链表。VMA 链表说明了特殊进程地址空间的映射关系。文件/proc/pid/maps 列出了不同 PID 号进程的 VMA 映射。当我们需要获得非 anonymous 页面的虚地址时，可以通过 address_space 和 VMA 结构找到对应的页表项。

虽然这种方法比直接查找反向页表要花费更多的时间，但是，因为可以不用维护非 anonymous 页面的反向页表结构，所以能够节省大量空间开销。我们考虑在今后的实现中将两种反向映射方法结合起来，并在不同的情况下使用不同的地址转换方法来获得更高的页迁移效率。

结束语 页迁移技术是实现 CC-NUMA 系统存储优化的重要方法，它动态地解决了数据局部性问题。由于迁移过程中频繁涉及到实虚地址的转换，其开销影响了页迁移的效率。以优化页迁移技术为出发点，本文提出了操作系统中实现快速实虚转换、支持页迁移的关键部件——反向页表技术。经过测试，在高端系统和负载很大的情况下，反向页表支持的页迁移系统性能明显优于传统页迁移系统。

当然，目前这种技术还面临着许多问题，主要体现在反向页表的空间开销上，而且 fork( )系统调用将会对进程地址空间中的每个页面增加一个新的反向页表项，大大增加了反向页表的维护开销。这些都是今后需要研究和解决的内容。

参 考 文 献

1 Verghese B, Devine S, Gupta A, Rosenblum M. Operating system Support for Improving Data Locality on CC-NUMA Computer Servers. In: proc. Architectural Support for Programming Languages and Operating Systems, 1996. 279～289
2 Nikolopoulos D, et al. Scheduler-Activated Dynamic Page Migration for Multiprogrammed DSM Multiprocessors. In Journal of Parallel and Distributed Computing, 2002
3 Laudon J, Lenoski D. The SGI Origin: A ccNUMA Highly Scalable Server[A]. In: Proc. of the 24th Annual Int'l Symp on Computer Architecture[C]. 1997
4 van Riel R. Towards an O(1) VM: Making Linux virtual memory management scale towards large amounts of physical memory. In Linux Symposium 2003
5 LWN The object-based reverse-mapping VM. http://lwn.net/Articles/23732/?format=printable

(上接第209页)

$\frac{1-0.6561}{1-0.9945327825}\approx62.9$倍；而使用完全复制的方法，只能将单点容错性能和整体容错性能分别提高$\frac{1-0.9}{1-0.99}=10$倍和$\frac{1-0.6561}{1-0.9509900499}\approx7.01$倍。由此可见，在增加同样的存储空间的情况下，环形 DCR²S 可以比完全复制更加有效地增加一组文件的单点和整体容错性能。

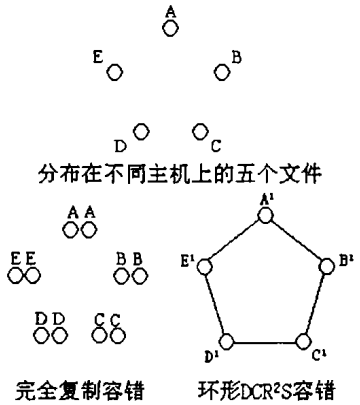

图6 环形 DCR²S 容错和完全复制容错

以上的分析是基于 $k=1$ 的假设，即环形 DCR²S 单元中每个点对应的文件复本只有一个。当 $k>1$ 时，根据公式计算可知环形 DCR²S 结构的容错性能仍然高于完全复制，且提高的幅度比 $k=1$ 时更大。由此可知，一组分散的文件在结成环形 DCR²S 单元后，若还想进一步提高单个文件和整体的容错性能，只需为每个文件增加复本，就可以获得比占用同样存储空间的完全复制方法更好的容错性能。

结论和未来的工作 本文提出了一种新的分布式协作冗余复制存储机制 DCR²S，这种机制通过校验文件实现了多个文件之间的相互协作，不仅可以提高单个文件的容错性能，更可以使得一组文件的整体容错性能得到大大提高。本文用图论的概念准确地描述了 DCR²S 的容错性质，定量地分析了链形和环形 DCR²S 结构的容错性能，并给出了这两种特殊结构的单点和整体可得概率的计算公式。通过计算比较，环形 DCR²S 结构在占用同样存储空间的条件下，可以实现比完全复制更高的容错性能，证明了 DCR²S 是一种十分有效的分布式文件容错机制。

如前所述，DCR²S 有多种结构，而本文只分析了链形和环形两种，在以后的工作中将对其他一些 DCR²S 结构的容错性能进行进一步的分析，找出容错效率最高、最易于实现的 DCR²S 单元结构，并通过实验给予验证。

参 考 文 献

1 Dingledine R, Freedman M, Molnar D. The freehaven project: Distributed anonymous storage service. In: Proc. of the Workshop on Design Issues in Anonymity and Unobservability, July 2000
2 Bolosky W, Douceur J, Ely D. Theimer M. Feasibility of a serverless distributed file system deployed on an existing set of desktop PCs. In: Proc. of Sigmetrics, June 2000
3 Kubiatowicz J. et al. Oceanstore: An architecture for global-scale persistent storage. In: Proc. of ASPLOS, Nov. 2000, ACM
4 魏青松，卢显良，雷宇. FTDSS：高容错分布式共享存储机制. 计算机科学，2003，30(8)：172～175
5 Druschel P, Rowstron A. Storage management and caching in PAST, a large-scale, persistent peerto-peer storage utility. In: Proc. of ACM SOSP (2001)
6 Blomer J, Kalfane M, Karp R, Karpinski M, Luby M, Zuckerman D. An xor-based erasure-resilient coding scheme. Technical report, Intl. Computer Science Institute, Berkeley, California, 1995
7 Weatherspoon H, Kubiatowicz J D, E C vs. Replication: A Quantitative Comparison, IPTPS '02