网格环境下基于编码机制的数据复制研究*)

陶 钧 沙基昌 王 晖

(国防科技大学信息系统与管理学院 长沙 410073)

摘 要 本文提出了基于编码机制的网格数据复制思想,通过对副本数据进行线性分组编码,并将其分散保存到网格存储节点,可形成具有纠删能力的编码子副本组。针对目前热点研究的线性分组编码,探讨基于 Cauchy Reed-Solomon Code、Tornado Code 和 Random Linear Code 的编码数据复制方案,通过建模手段讨论三者的副本数据访问性能和副本数据可靠性,并与传统的完整数据复制和分块数据复制进行对比分析,证明所提出的编码数据复制有着较优的综合性能。具体实验数据进一步说明,编码副本的编码开销占整个数据复制开销的较小比例,表明编码数据复制是具有可行性的技术方案。

关键词 网格,数据复制,线性分组编码,编码副本

Research on the Coding Mechanism-based Data Replication in Grid Environments

TAO Jun SHA Ji-Chang WANG Hui

(National University of Defense Technology, School of Information System and Technology, Changsha 410073)

Abstract This paper proposes a thought of coding mechanism-based grid data replication. By carring on the linear block coding to the replica data and dispersedly saving them to the grid storage nodes, erasure capacity can be formed with the coded child-replica group. Focusing on the current hot linear block coding, this paper presents the schemes of the coding data replication based on the Cauchy Reed-Solomon code, Tornado Code and Random Linear Code, and discusses the data access and reliability performance of the replicas by the modelling method. Compared with the complete data replication and block data replication, the coding data replication has the overall better performance. Specific experimental data further illustrate that the coding expenses of the coded child-replica account for a relatively small proportion of the data replication, and indicate that the coding data replication is a feasible technical scheme.

Keywords Grid, Data replication, Linear block coding, Coded replica

1 引言

数据复制被认为是提高网格数据访问性能和容错性能的重要技术途径,它能有效减少用户到存储节点的数据访问延迟和带宽消耗,同时减轻存储节点的服务压力,避免因存储节点故障引起的数据单点失效现象。在网格环境下,数据复制概念就是产生原始数据的一个或多个内容拷贝,将它们存储到地理分散且错误独立的存储节点。随着网格数据规模的不断扩大,如何高效地存储这些副本数据是研究人员所面临的挑战。

近年来,分布式存储技术取得了长足的发展,充分利用网络资源实现高效的虚拟化存储成为研究的重点。特别是基于容错编码理论的分布式编码存储的产生,如 Data-Harvesting[1]、OceanStore[2]、Myriad[3],通过数据的分块编码冗余以及存储节点间的联合协作,实现了分布式条件下的数据高可靠性存储。分布式编码存储的设计成功,促使人们进一步探讨编码机制在分布式存储领域中的可能应用。

根据上述研究现状,本文提出了基于编码机制的网格数据复制思想,尝试利用线性分组编码后的副本数据,设计适应网格环境的数据复制方案,并将其与传统的完整副本和分块副本的网格数据复制进行比较,同时研究不同线性分组编码策略下的副本数据可靠性和副本编码效率。

本文第 2 节讨论编码机制的数据复制设计思路。第 3 节研究基于编码机制的数据复制整体方案。第 4、5 节着重对比分析编码数据复制与传统数据复制的性能差异,并通过实验论证编码数据复制的可行性。最后总结全文。

2 编码机制的数据复制思想

分布式编码存储[1~4]是通过将编码后的数据切片存放到地理分散的多个存储节点,以实现分布式环境下的存储容错,提高整个系统的可靠性和可扩展性。其采用的编码策略通常为线性分组编码[4],编码方式则以代数形式和随机形式为主。

线性分组编码的简单高效特性,使其适应于数据密集型的网格数据复制环境。文[5]对编码机制的数据复制进行了探索,文[6]则设计了运用代数形式编码的数据复制协议,文[7]进一步探讨编码数据复制于传统数据复制的优势,并对代数与随机形式的编码方案进行比较。目前热点研究的线性分组编码包括:Linear Erasure Code[8~11] 和 Random Linear Code[12,13],基于上述两类分组编码策略,将其引入到网格副本的生成,以提高网格环境下数据复制的综合性能。

2.1 基于 Linear Erasure Code 的数据复制思想

Linear Erasure Code 是基于代数生成规则的线性分组编

*)本论文受国家防预研基金支持。陶 钧 博士研究生,主要研究领域为计算机网络科学、信息系统管理科学;沙基昌 教授,博士生导师,主要研究领域为信息系统管理科学、军事运筹学;王 晖 博士,教授,主要研究领域为计算机网络科学、信息系统管理科学。

码方案。它作为传统的纠删编码技术，曾广泛运用于各类通信系统[8]和冗余磁盘阵列(RAID)的数据存储[15]领域。Linear Erasure Code 的编码是将数据均分为 m 个数据块，根据特定的代数规则形成 $m\times n$ 的编码矩阵，以此生成 $n(n>m)$ 个大小不变的编码数据块。解码时，获取 n 个编码数据块中的任意 $r(r\geqslant m)$ 个，就可解码重构出原始数据。Linear Erasure Code 存在许多优化改进方案，其中应用最为广泛的分别是 Cauchy Reed-Solomon Code 和 Tornado Code。

(1)Cauchy Reed-Solomon Code(CRS)[8]被认为是目前最优秀的 MDS 编码[9]，满足解码所需数据块 $r=m$。

(2)Tornado Code[10]是一种稀疏系数的快速编码方案，该方案使得编解码复杂度大大降低；在 m 较大的情况下，编解码速度是一般方案的几百甚至上千倍[11]。

基于 Linear Erasure Code 的特点，其编码数据复制的思想为：网格数据的一个副本由 $n(n$ 值固定)个编码子副本组构成，数据冗余度达到 n/m，网格复制目录将记录这些子副本所对应的存储节点信息。当用户访问副本时，先从网格复制目录获取子副本对应的节点信息，再访问其中 r 个性能优秀的子副本，以解码重构出副本数据。

2.2 基于 Random Linear Code 的数据复制思想

Random Linear Code 是基于随机生成规则的线性分组编码方案。它最早产生于密码学应用领域，在网络编码[12,13]的研究中被提出用来解决随机流问题，文[7,13]则以此为基础将其引入到数据存储范畴。Random Linear Code 的编码是在有限域 GF(q)上随机生成 $n(n\geqslant m)$ 组 m 维向量，再将原始数据均分为 m 块后以 m 维向量为系数进行有限域运算，从而获得 n 个大小不变的编码数据块，其编码公式如下：

$$F_i=\sum_{j=1}^{m}\beta_j D_j \quad \Pr(\beta_i=\beta)=1/q,\ \forall\beta\in GF_q \quad (1\leqslant i\leqslant m) \tag{1}$$

其中 D_j 为分割后的原始数据块，β_j 为有限域[0,…,$q-1$]中的随机值，F_i 为向量($\beta_1,\beta_2,\cdots,\beta_m$)所对应的编码数据块，$Pr$ 为向量 β_j 的生成概率。解码时获取符合高斯消元要求的任意 m 个编码数据块，满足对应编码向量线性无关，即可解码重构出原始数据。实验证明，以随机理论为基础的 Random Linear Code 能够适应于大规模分布式应用环境[7]。

基于 Random Linear Code 的特点，其编码数据复制的思想为：网格数据的一个副本由 $n(n\geqslant m$ 的任意值)个随机编码子副本组构成，数据冗余度 n/m 可随需求而改变，网格复制目录不光记录子副本对应的存储节点信息，还要同时保存生成子副本的随机编码向量。当用户访问副本时，要同时从网格复制目录获取子副本所对应的节点信息和编码向量，再访问其中 m 个编码向量线性无关的子副本，以解码重构出副本数据。

3 编码机制的数据复制方案设计

考虑典型的层次化拓扑结构网格环境，并且根据网格发展的趋势，允许网格内域间的扁平化数据访问；域内存在许多网格存储节点提供服务；中心域具有全局的复制目录。

3.1 编码机制的数据复制整体分析

通常来说，分布式系统与集中式系统的重要区别就在于数据的分散性，由此带来的数据一致性问题，始终是分布式系统所要关注的内容。文[7]就指出分布式编码存储并不适合于数据频繁更新的应用环境。而对于网格环境的数据复制而言，可看作是原始数据特定时刻的快照(Snapshot)，并对其进行长期缓存(Cache)的过程，那么网格副本是具有时效性的。

通过上面的分析可以看出，作为传统网格数据复制的扩展形式，编码数据复制的子副本具有阶段性只读特征，其分散存储并不会引起高并发的数据一致性问题。简单的一致性管理方案就是在分散存储子副本的同时，以携带版本编号的方式记录到网格复制目录。图 1 描绘出编码数据复制的逻辑示意图。

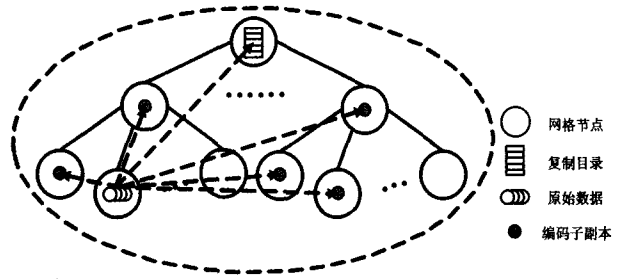

图 1 编码数据复制的逻辑示意图

3.2 编码机制的数据复制功能设计

通常来说，完整的网格数据复制应当包括副本写操作、副本读操作、副本更新操作和副本回收操作，它们是构成数据复制的基本功能模块。

(1)副本写操作：编码副本的写操作可分为副本数据编码和副本分散存储两个阶段。副本数据编码阶段是对原始数据进行分组编码的过程，形成预期冗余度的编码子副本组；副本分散存储阶段可采用文[14]中的存储节点选择策略，实现自适应的子副本存储。

(2)副本读操作：编码副本的读操作可分为副本数据访问和副本解码重构两个阶段。副本数据访问阶段是对多个子副本进行并行访问的过程，这将提高副本数据的整体访问效率[7]；副本解码重构阶段是多个编码子副本进行数据解码并对解码数据块组合还原的过程[8~11]。

(3)副本更新操作：分散存储的编码子副本并不随原始数据的更新而实时更新，通过定期比对网格复制目录中副本版本与原始数据版本是否一致，来决定是否重新生成最新版本的编码子副本组，也就是说副本更新操作是简单的重新生成方式。

(4)副本回收操作：由于副本采用重新生成的版本更新方式，因此通过定期的版本检查机制，可以删除低版本的子副本数据，从而回收存储节点及复制目录上的存储空间。

4 编码机制的数据复制性能分析

编码数据复制不同于传统的网格数据复制方式，本节将以模型化的形式研究编码数据复制的综合性能。为了简化问题的分析复杂度，对网格环境做出了合理假设：存储节点失效概率相等且失效相互独立；全局的复制目录总可以访问。

4.1 传统网格数据复制的分类

传统的网格数据复制可分为完整数据复制和分块数据复制两类。完整数据复制是拷贝原始数据的全部内容，在单个存储节点上保存完整副本。分块数据复制是将原始数据均分为多个数据块(但不进行编码)并分散保存到多个存储节点，用户通过访问分块副本可以重构出原始数据。图 2 对比显示了完整数据复制、分块数据复制与编码数据复制的原理结构。

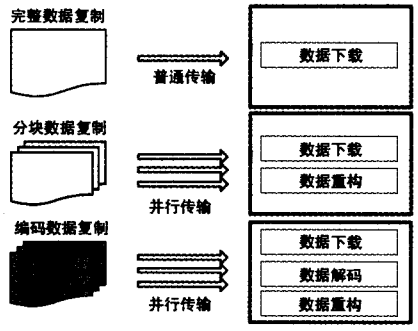

图 2　三种数据复制方式的对比

4.2　传统数据复制的性能分析

(1)数据分散对副本访问性能的影响

副本数据的访问性能是副本获取过程中所产生的访问延迟。当副本数据量较大时,访问延时主要是副本的传输时间。设副本数据量为 L,存储节点允许的下载带宽上限为 M_D,客户机的下载带宽上限为 M_U,网格内通信的瓶颈带宽为 M_L。考察完整数据复制和分块数据复制的传输延时。

对于完整数据复制,访问延时 T_{R1} 在理想情况下可公式化表示为:

$$T_{R1}=L/\mathrm{Min}(M_D,M_U,M_L) \tag{2}$$

对于分块数据复制,访问延时 T_{R2} 在理想情况下可公式化表示为(副本分块数量为 m)

$$T_{R2}=L/\mathrm{Min}(mM_D,M_U,M_L) \tag{3}$$

当然,实际传输延时不可能达到理想情况,但是从式(2)、(3)可以看出 $T_{R1}\geqslant T_{R2}$,这说明由数据分散保存而产生的并行传输方式,能一定程度提升副本数据的访问性能。

(2)数据分散对副本可靠性的影响

分块数据复制的数据分散机制会对副本的可靠性产生影响。设单个存储节点的可靠性为 $P(0<P<1)$,副本数据的冗余度为 γ。考察完整数据复制和分块数据复制的副本可靠性。

对于完整数据复制,副本数据可靠性 P_{R1} 可公式化表示为

$$P_{R1}=1-(1-P)^{\gamma} \tag{4}$$

对于分块数据复制,副本数据可靠性 P_{R2} 可公式化表示为(副本分块数量为 m)

$$P_{R2}=(1-(1-P)^{\gamma})^{m} \tag{5}$$

比较式(4)、(5),可以发现在相同副本冗余度时,副本数据可靠性 $P_{R1}>P_{R2}$,并且随着分块数量 m 的增加,P_{R2} 单调递减。

4.3　编码数据复制的性能分析

从 4.2 节的对比分析可以看出,完整数据复制的副本访问效率较低;而分块数据复制为提高访问性能,却牺牲了副本数据的可靠性。那么对于编码数据复制而言,采用与分块数据复制相同的数据分散方式,能获得较好的访问性能。下面则对编码数据复制的可靠性进行建模分析。

(1)基于 Linear Erasure Code 的编码副本可靠性

基于 Cauchy Reed-Solomon Code 编码方案的数据复制(对于 Tornado Code 则可认为略低于 Cauchy Reed-Solomon Code 的可靠性[11]),是将编码后的 n 个编码子副本存放到 n 个存储节点上。由于发生 0 至 $n-m$ 个子副本失效,子副本组均能完成副本数据重构,那么副本数据可靠性 P_{R3} 可公式化表示为

$$P_{R3}=\sum_{i=0}^{n-m}\binom{n}{i}P^{n-i}(1-P)^{i} \tag{6}$$

(2)基于 Random Linear Code 的编码副本可靠性

基于 Random Linear Code 编码方案的数据复制,是将随机编码后的 n 个子副本存放到 n 个存储节点,根据有限域随机行列式秩的计算公式,有

$$P_M(d,m,n)=\prod_{i=0}^{d-1}(q^n-q^i)\prod_{i=0}^{d-1}(q^m-q^i)/q^{mn}\prod_{i=0}^{d-1}(q^d-q^i) \tag{7}$$

式(7)中 P_M 为行列式秩为 d 的概率,m 为行向量个数,n 为列向量个数,q 为有限域大小。由于解码算法要满足 m 维的高斯消元要求,那么副本数据可靠性 P_{R4} 可公式化表示为

$$P_{R4}=\sum_{i=0}^{n-m}\binom{n}{i}P^{n-i}(1-P)^{i}P_M(m,m,n-i) \tag{8}$$

编码数据复制的副本冗余度为 $\gamma=n/m$,那么比较分析可靠性公式(4)、(5)、(6)、(8)可以发现,总满足关系式 $P_{R3}>P_{R4}>P_{R2}$ 成立;而且当存储节点的可靠性 P 较高、子副本数量 n 较大时,有关系式 $P_{R3}>P_{R4}>P_{R1}$ 成立。

由此可以认为,编码数据复制在副本访问性能和副本可靠性方面性能较优,与传统数据复制相比具有一定的优势。

5　编码机制的数据复制可行性分析实验

编码数据复制与传统数据复制的重要区别在于需对副本数据进行线性分组编码,这会带来可预见的计算资源开销,这也是编码数据复制较传统数据复制相比的不足之处。本节对编码数据复制的编码效率进行分析。

5.1　编码效率的理论分析

从理论角度对子副本的编解码计算效率进行分析:Cauchy Reed-Solomon Code 可以采用系统码方式[8],因此只需编码 $n-m$ 个子副本;解码时可获得部分系统码子副本,直接降低解码计算复杂度。Tornado Code 采用二分图编码生成方式,因此编码系数十分稀疏,只有普通分组编码密度的 1/m[11]。Random Linear Code 产生 n 个 m 维随机向量[13],因此编解码需进行 n 次 m 维行列式运算。根据以上讨论分析,表 2 归纳出编码数据复制的编码效率理论分析情况(L 为副本数据量)。

表 2　不同编码方案的编码效率比较

	Cauchy Reed-Solomon Code	Tornado Code	Random Linear Code
编码时间	$O((n-m)L)$	$O(nL/m)$	$O(nL)$
解码时间	$O(m(n-m)L/n)$	$O(nL/m)$	$O(mL)$

5.2　编码效率的比较实验

编码数据复制由副本编码和副本传输两部分组成,那么分别实验计算副本编码开销与传输开销,考察副本编码对整个数据复制过程的影响。

(1)副本数据的编解码实验

测试平台为 P4 1.8G(512MB,Linux)台式机,测试代码改造于文[9]的 C 语言参考程序。实验过程中取副本冗余度为 $\gamma=2$,改变副本大小和分块数量进行实验,观察编解码时间的变化。

表 3 不同编码方案的编码效率比较

副本大小(MB)	m	n	Cauchy Reed-Solomon Code 编码时间(s)	Cauchy Reed-Solomon Code 编码时间(s)	Tornado Code 编码时间(s)	Tornado Code 编码时间(s)	Random Linear Code ($q=2^8$) 编码时间(s)	Random Linear Code ($q=2^8$) 编码时间(s)
10	4	8	0.485	0.233	0.213	0.105	0.901	0.242
20	4	8	1.197	0.552	0.530	0.252	2.401	0.561
30	4	8	1.582	0.776	0.734	0.346	3.242	0.798
40	4	8	2.254	1.012	0.989	0.498	4.095	1.021
10	8	16	0.645	0.312	0.245	0.120	1.253	0.320
20	8	16	1.466	0.701	0.572	0.257	2.877	0.782
30	8	16	2.008	1.022	0.806	0.363	4.102	1.084
40	8	16	2.639	1.714	1.051	0.478	5.242	1.832

实验采用的是 10-40MB 的副本数据，对于更大甚至上 G 级的副本而言，其编解码也是划分为 M 级的数据段进行的，因此大副本数据的编解码也符合相同的线性增长规律。

(2)副本数据的网络传输实验

测试平台为五台 P4 1.8G(512MB,Linux)台式机，GridFTP 版本为 2.1，网络环境为 100M 局域网。实验过程中 TCP Buffer 为 64kb，改变传输副本大小和传输模式进行实验，观察网格数据传输时间的变化。

表 4 不同传输模式的传输效率比较

副本大小(MB)	Parallelism=0 Striping=0 传输时间(s)	Parallelism=4 Striping=0 传输时间(s)	Parallelism=4 Striping=2 传输时间(s)	Parallelism=4 Striping=4 传输时间(s)
10	3.062	3.185	1.893	1.584
20	6.754	6.511	3.847	2.827
30	10.117	9.962	5.016	4.182
40	13.298	13.141	6.772	5.554

实验是在封闭网络条件下进行的，可以认为是同等条件的最大传输效率。对于第四种传输模式，则基本上达到了客户机的网络接受上限。

通过对比表 3 和表 4 的实验结果可以发现：即使是在最优的传输模式下，对相同大小的副本数据来说，编码时间仍接近于传输时间，解码时间则远低于传输时间。如此的编码开销与传输开销比值是可以接受的，因为可采用数据编码与数据传输同步进行的方式，使得编码开销只是消耗整个数据复制过程的计算资源，而几乎不消耗时间资源。那么即使是处理上 G 级的副本数据，相比于传统数据复制过程来说，编码数据复制将也不会有明显的操作时间增加。

总结 本文提出了基于线性分组编码机制的网格数据复制思想，通过对 Cauchy Reed-Solomon Code、Tornado Code 和 Random Linear Code 进行建模比较分析，证明基于编码机制的数据复制较传统数据复制有着传输性能和可靠性方面的优势，并通过实验证明副本编码开销与传输开销的比值在可接受范围内。本文研究表明，编码机制的数据复制具有很强的实际应用价值，是值得进一步深入研究的网格复制技术方案。

今后的工作重点将会是编码机制的数据复制管理研究，传统的复制管理对于编码数据复制而言有其局限性，改进复制管理策略将是编码数据复制实用化的关键一环。

参 考 文 献

1 Deb S, Choutte C, M'edard M, et al. Data harvesting: A random coding approach to rapid dissemination and efficient storage of data:[M. I. T. LIDS Technical Report]. 2004
2 Kubiatowicz J, et al. Oceanstore: An architecture for global-scale persistent storage. In: Proceedings of ASPLOS, 2000
3 Chang F, et al. Myriad: Cost-effective Disaster Tolerance. In: Proceedings of FAST, 2002
4 Frolund S, Merchant A, Saito Y, et al. A decentralized algorithm for erasure-coded virtual disks. In: Proceedings of DSN, 2004
5 Weatherspoon H, Kubiatowicz J D. Erasure Coding vs Replication: A Quantitative Comparison. In:Peer-to-Peer Systems: First International Workshop, IPTPS 2002, LNCS2429, 2002
6 Zhang Z, Lian Q. Reperasure: Replication Protocol Using Erasure-code in Peer-to-Peer Storage Network. In: Proc. of the 21st IEEE Symp. on Reliable Distributed Systems (SRDS 2002), 2002
7 Acedanski S, Deb S, Medard M, et al. How Good is Random Linear Coding Based Distributed Networked Storage. In:Proceedings of First Workshop on Network Coding, 2005
8 Luby M, Mitzenmacher M, Shokrollahi A, et al. Practical loss-resilient codes. In: 29th Annual ACM Symposium on Theory of Computing, ACM, 1997
9 http://www.icsi.berkeley.edu/~luby/erasure.html
10 Byers J W, Luby M, Mitzenmacher M. Accessing multiple mirror sites in parallel: Using tornado codes to speed up downloads. In: IEEE INFOCOM, New York, 1999
11 Byers J W, Luby M, Mitzenmacher M, et al. A Digital Fountain Approach to Reliable Distribution of Bulk Data. In: Proceedings of ACM Sigcomm '98, Canada, 1998
12 Li S Y R, Yeung R W, Cai N. Linear network coding. IEEE Transactions on Information Theory, Februray 2003
13 Gkantsidis C, Rodriguez P. Network coding for large scale content distribution: [Technical Report MSR-TR-2004-80]. Microsoft Research, 2004
14 Lee Byoung-Dai, Weissman J B. An Adaptive Service Grid Architecture Using Dynamic Replica Management. GRID, 2001. 63～74
15 Chen P, et al. RAID: High-Performance, Reliable Secondary Storage [A]. ACM Computing Surveys, 1994

(上接第 100 页)

由表 1 可知 k 值对实际攻击的侦测率不会造成影响，但是会影响错误警告，因为当 K=3 时，子网络内的某台主机遭受攻击会连带影响其它主机的封包也一并被丢弃，造成错误警告增加。由表 2 得知 n 值越小越能快速侦测到攻击发生，但相对的错误警告会增加。借由提升 Rmax，我们可以消除错误警告的负面效果。在 n=50、Rmax=6 条件下会产生 750 个错误警告，因为当时的流量分布是 8 个 http 封包才有一个 ack 封包，又加上 n=50 使得快速累积到攻击门槛。由表 4 得知 α 值对侦测率与错误警告的影响很小，随着 α 值变大，错误警告会降低但侦测率也相对下降。

由以上的各种实验，得出一个重要结论，想要快速侦测攻击发生则 n 值越小越好，但相对的误判机率就会升高，此时需将 Rmax 值调高来减少误判。我们建议各个参数值的设定为 $K=4, n=50, Rmax\geqslant 8, \alpha\leqslant 10$。

小结 我们在阻断服务攻击侦测方面，结合 TOPS 的储存架构与连续假设实验的概念，使得平均侦测效果比原本的 TOPS 还好，演算法的实现与运算上也更容易。未来希望可以持续改进演算法，将演算法实现在硬件上，借由实际的封包测试来看整体的表现。

参 考 文 献

1 陈晶，崔国华，洪亮，付才. 一种 Ad Hoc 网络中的安全匿名按需路由协议. 计算机科学，2007,34(1)
2 陈云开，孙小林，马君华. 外汇领域的洗钱侦测系统及关键算法研究. 计算机科学，2007,34(3)
3 Jung J, Paxson V, Berger A W, et al. Fast Portscan Detection Using Sequential Hypothesis Testing
4 [美]Cole E 著.《黑客——攻击透析与防范》. 电子工业出版社，2002.2