流媒体服务中即时响应的层次存储数据放置策略*)

徐尧强　邢春晓　周立柱
(清华大学计算机科学与技术系　北京100084)

摘　要　为了获得较高的性能价格比，流媒体服务器通常采用层次存储技术。由于三级存储设备的机械特性，访问层次存储系统(HSM)中的数据需要很长的响应时间。本文提出了具有即时响应性能的数据放置方法：把流媒体对象特定长度的头部数据预先放置在磁盘上，而数据请求首先从HSM的磁盘上得到响应，在读取磁盘数据期间进行三级存储设备的准备。在特定的头部数据长度以及读取算法下，可以消除访问三级存储设备需要的等待时间。本文给出了此头部数据长度的计算方法以及HSM的数据读取算法。仿真试验表明，只需较小的存储代价，就可以使得HSM系统的响应时间大大降低，而且数据具有良好的连贯性，从而提高了整个系统的性能。

关键词　层次存储，媒体服务器，数据放置

An Immediately Responsible Data Placement Method of Hierarchical Storage Management for Streaming Media Server

XU Yao-Qiang　XING Chun-Xiao　ZHOU Li-Zhu
(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

Abstract　Hierarchical storage technology is used in media servers to get high performance/cost ratio. However, in traditional hierarchical storage system (HSM), the data, say a movie, is usually placed wholly in a tertiary storage device. Thus, the response time to user's request will be too long to be satisfactory. This paper presents a new data placement method based on prearrangement strategy in HSM for streaming media server. By prearranging the head part of media objects with specified length on disk, streaming data accesses can get immediate response, and in the meantime, tertiary storage device can be ready for continuous access. The paper gives a method to calculate the head data length of media objects and an algorithm to read data in HSM. Simulation result shows that with a low disk storage cost, the response time of the HSM system can be reduced to a large extent, and the performance of the whole system is enhanced significantly.

Keywords　Hierarchical storage, Streaming media server, Data placement

1　引言

随着网络与计算机技术的发展，流媒体服务(如视频点播，新闻点播等)已得到日益广泛的应用。大型媒体服务器拥有上千个媒体对象，全部存储在磁盘上时需要很大的代价，而其中部分媒体对象却很少被访问。为了取得较高的性能价格比，很多文献提出采用分布式层次存储[1,2]，把较少访问的媒体对象存储于层次存储系统(Hierarchical Storage Manager，HSM)中。HSM是指把磁盘等性能较高的二级存储设备与三级存储设备(如磁带库、光盘塔等)组合在一起，拥有适中性能与代价的存储系统。使用这种存储系统的优点是存储代价较低，但缺点是访问时需要用机械臂交换介质，并进行定位，通常有几十秒或几分钟的等待时间。

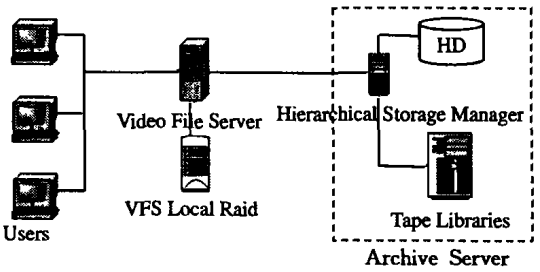

图1　基于层次存储的媒体服务系统

图1是一种典型的基于层次存储的媒体服务器结构。媒体文件服务器(Video File Server，VFS)接受用户的点播请求并提供服务，它拥有高速磁盘阵列，存储少量经常被访问的媒体对象，其他的媒体对象存储于HSM中。当用户请求的媒体对象不在VFS上时，首先将数据从HSM读到VFS，再由VFS提供服务。

目前HSM领域的研究主要针对科学计算型数据或通用数据，往往把绝大部分的数据全部放置在三级存储设备上，其研究目标是减小读取整个数据的代价或介质的交换次数[3,4]，不能有效地缩短访问HSM系统的响应时间。文[5]提出媒体服务器中的一种流水线机制：当请求的媒体对象不在VFS上时，从HSM向VFS读取数据，当VFS中具有一定的数据后就开始向用户提供服务，此后采用流水线机制，一面从HSM读取数据，一面向用户提供数据。这种机制避免了文件拷贝过程的等待，但由于在开始阶段仍然需要把磁带库的数据读入磁盘，因此仍然不能消除由此产生的延迟。

本文提出基于预置的流媒体数据放置策略，在面向媒体服务的HSM中，把媒体对象一定长度的头部数据预先放置在磁盘上，后续数据存储于三级存储设备中；访问HSM时首先从磁盘获得响应，同时准备三级存储设备。当磁盘上的数据长度合适时，三级存储设备将在读取磁盘数据的过程中准备

*)基金项目：国家重点基础研究发展规划资助项目(G1999032704)。徐尧强　博士生，主要研究领域为海量信息处理；邢春晓　博士，副教授，主要研究领域为海量信息处理，数据库技术和数字图书馆；周立柱　教授，博士生导师，主要研究领域为数据库，海量信息处理，Web技术。

完毕，从而在读完磁盘数据时，可以连贯地过渡到从三级存储设备读取数据。这种策略使得从 HSM 读取数据的等待时间接近于直接从磁盘读取数据的响应时间。预置数据的思想在文[6]中也有提及，但是方法太简单，不具有即时响应性能，而且它没有解决数据的连贯性问题。

本文第2节讨论三级存储设备的访问特性和流媒体服务的主要特点。第3节介绍基于预置的流媒体数据放置策略，并讨论预置于磁盘上的头部数据长度的计算方法，还给出了 HSM 中读取媒体对象的算法。第4节是仿真试验结果和性能分析，最后是总结。

2 三级存储设备的访问特性和流媒体服务的基本特点

HSM 主要采用三级存储设备来存储数据，同时采用少量的磁盘以提升性能。三级存储设备主要指磁带库和光盘塔。前者采用磁带作为存储介质，拥有较大的容量；后者采用随机访问的光盘为存储介质，容量较小，但性能较高。

三级存储设备通常拥有一个或几个驱动器，以及很多个用来放置存储介质的槽，并有一个(很少多于一个)机械臂。存储介质，如磁带或光盘，平时放在槽上。使用时，由机械臂把介质从槽上取到驱动器，由驱动器读写数据。使用结束后，仍然由机械臂把介质从驱动器中取出放回槽中。因此，访问三级存储设备中的数据时，其响应时间为：

$$T\_reponse = T\_load + T\_mount + T\_seek \quad (1)$$

其中，T_load 指机械臂把介质从槽转移到驱动器需要的时间，通常在10秒左右。T_mount 指把介质放入驱动器到驱动器可以操作介质需要的时间。T_seek 指驱动器定位到目标位置所需的时间。对于光盘 T_seek 很短，而对于磁带，根据性能和目标位置不同，从几十秒到几百秒不等。

当介质准备好之后，三级存储设备可以以较高的速度持续读取数据，而媒体服务，一方面要求较短的响应时间，另一方面也要求数据到达必须保证一定的速率，否则将导致媒体服务质量的下降。但是，媒体服务所需要的数据读取速率并不高。以 MPEG2 格式的视频流为例，其码率(bit rate)在 4.5Mbps 左右，即0.6Mbyte/s。

表1 存储设备性能

介质	响应时间(s)	数据读取速率(MBps)
Sony DTF 磁带库	143	12
Ampex DST 磁带库	25	14.2
IDE 硬盘	0	8

表1列出了两种磁带库以及普通 IDE 硬盘读取数据的性能指标，Sony DTF 数据来自文[7]，Ampex DST 数据来自文[4]。从表中可以看出，存储设备的持续数据读写速度远远大于媒体播放速率，但是在响应时间上，只有二级存储设备能较好地满足媒体服务的需要。

媒体服务的另一个特点是：在很多情况下，大量的访问集中于少数媒体对象。这种访问大致基于 Zipf(0)分布储[1,8]，即 M 个媒体对象中，第 n 个最流行的媒体对象的访问概率是 C/n，其中 $C=1/(1+1/2+1/3+\cdots+1/M)$。

3 基于预置的流媒体数据放置策略

本文描述的策略主要针对目前最通用的磁带库。这种磁带库使用螺旋扫描的磁带，回卷时磁头定位到磁带头部。其思想也可直接用于光盘塔或其他类型的磁带库。

3.1 基于预置的流媒体数据放置策略

定义1(文件，Bit File) HSM 系统中连续的字节流，对应于存放到系统中的媒体对象。

定义2(段，Segment) 文件由若干段组成，其中的每一段是连续的数据，是文件的物理存储单位。

定义3(文件分段信息) 文件和段的对应关系，即一个文件由哪几个段组成。

媒体对象以文件的形式存放在 HSM 系统中。每个文件都分成两个段，如图2所示。头部长为 L_head_seg 的段存储于 HSM 系统的磁盘，余下数据存储于 HSM 系统中的磁带上。为了保证磁带库能在读取磁盘数据的过程中准备完毕，L_head_seg 应该满足下面的公式：

$$L\_head\_seg = V\_bit\_rate * (T\_response + T\_queue) \quad (2)$$

其中 V_bit_rate 是媒体对象播放时的码率，$T\_response$ 指从磁带库读取数据需要的延迟，而 T_queue 是用于衡量等待使用三级存储设备的排队时间，下一节将讨论 T_queue 的计算方法。

忽略启动过程，磁带的定位时间可如下计算：

$$T\_seek = \frac{S\_addr}{V\_seek} \quad (3)$$

其中 S_addr 指目标在磁带上的位置，V_seek 指磁带快速定位的速率。

综合公式1、2和3，可得：

$$L\_head\_seg = V\_bit\_rate * (T\_load + T\_mount + \frac{S\_addr}{V\_seek} + T\_queue \quad (4)$$

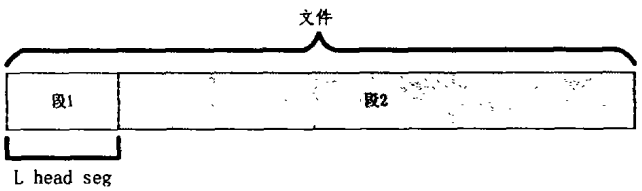

图2 媒体文件分段

3.2 T_queue 的计算

数据放置策略中，T_queue 是很重要的一个参数。但是由于系统的复杂性，没有很好的解析方法可用于根据系统负载和配置等因素来求解。我们通过仿真手段，统计不同系统负载下请求的延误时间，进而根据延误时间的分布确定合适的 $T\_overload$。

媒体服务器的排队模型描述如图3所示。

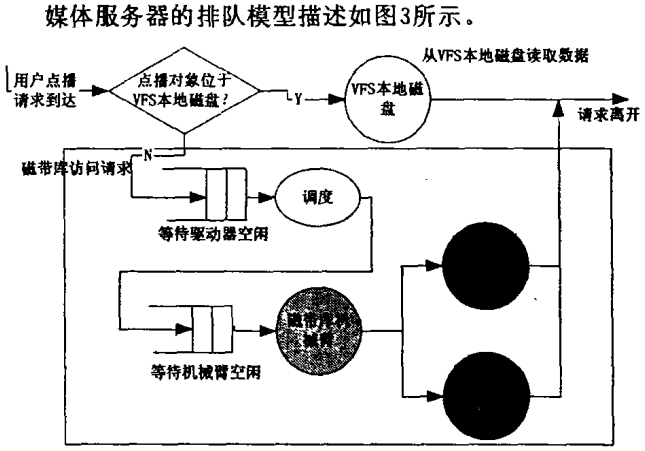

图3 媒体服务器排队模型

如果用户请求的媒体对象存在于 VFS 本地磁盘，则从中读取数据并立即提供服务，否则从 HSM 获取数据：首先进入调度等待队列，直到有驱动器空闲；然后调用机械臂把磁带 load 到相应驱动器中，定位并读取数据；最后退出磁带并由机械臂放回槽中(未在图中画出)。用户对媒体服务器的点播请求一般基于泊松分布(请求强度的变化比较缓慢)，访问的媒体对象则服从 Zipf(0)分布。令每个磁带驱动器单位时间内能处理的用户请求个数为驱动器的处理能力 μ，而单位时间内到达 HSM 系统的用户请求的个数为请求强度 λ，而系统中驱动器个数为 K。定义 HSM 系统的负荷水平为：

$$\rho(K)=\frac{\lambda}{K*\mu} \tag{5}$$

λ 可以通过用户访问媒体服务器的请求强度和请求的对象不在 VFS 上的概率得到。

我们基于 Sim++[9,10]，在不同负荷水平下，对从 HSM 系统读取数据的请求的等待时间进行了仿真试验。系统中采用 Sony DTF 驱动器，机械臂指标依照 Sony DMS-B9。具体参数如表2所示(数据来源于文[6])。

表2 系统主要参数

符号	数值	说明
Drive_count	4	磁带库中驱动器的个数
TapeLoad_time	12.8s	机械臂把带从槽中放到驱动器需要的时间
TapeUnload_time	17.2s	机械臂把带从驱动器中取出放到槽中需要的时间
TapeMount_time	51s	从磁带塞到驱动器中到可以操作磁带的时间
TapeUnmount_time	18s	从发出 unmount 命令到机械臂可以从驱动器取磁带的时间
Transfer_rate	12Mbyte/s	磁带持续读写速度
Seek_rate	300Mbyte/s	磁带快速定位的速度
Tape_size	42G	磁带容量
Object_size	2G	每个媒体对象的大小。这里我们认为 Object-size 是一致的
Tape_count	20	磁带库中的磁带总数(这个可以由系统需要存储的数据决定)

图4显示了不同负荷下，从 HSM 读取数据的请求的等待时间分布。曲线上一点 (t,p) 表示在这种负载下，等待时间不大于 t 的请求占总请求数的比例为 p。公式5求出的负荷并不能唯一衡量系统的运行，同样的负荷下，驱动器越多，系统响应性能越好，而这也是符合随机过程的基本原理的。

可以看出，大部分请求的等待时间都在不太大的间隔内。参照分布曲线，我们可以根据系统的需要，选择一个合适的域值作为 T_queue，使得大部分请求都可以在这个等待时间内得到处理，而付出的存储代价也控制在合适的范围内。

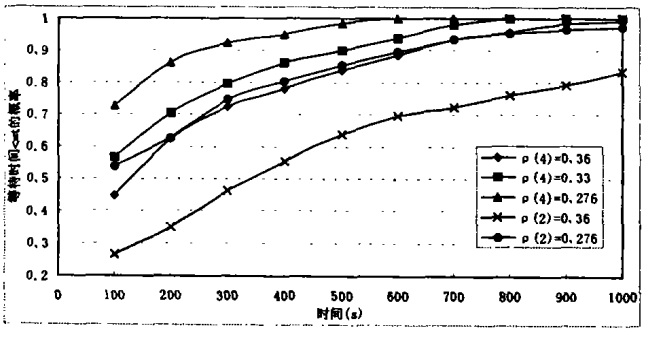

图4 不同负荷下 HSM 请求的等待时间分布

3.3 数据读取算法

按照上述数据放置策略预置媒体对象后，为了充分利用其优点获得尽可能短的响应时间，需要有特定的数据读取算法。同时，预置的头部数据的长度 L_head_seg 只能保证大部分的请求能即时响应，实际使用中可能出现实际排队时间超过预期的情况，此时需要在数据读取算法中保证数据的连贯性。

为了保证数据流的连续性，HSM 在收到一个请求时，首先必须预测此请求的执行时间，为此磁带库请求的调度需要采用先来先服务的策略。HSM 在接收到一个请求时，首先根据任务队列确定这个请求需要的实际等待时间。如果超过了头部数据所能维持的范围，则首先进行等待同时通知用户需要等待的时间，等待结束后从磁盘开始读取数据。

在实际的 HSM 中有多个线程(进程)处理磁带库的读写请求，为清晰起见，这里只给出主控线程的数据读取算法描述：

①接受媒体服务器读取文件的请求；

②查询文件分段信息；

③把读取文件第二段的请求放入磁带库等待队列，并根据等待队列计算读取文件第二段需要的等待时间 T_wait；

④如果 $T\_wait <= L\_head\_seg / V\_bit\_rate$，则转5，否则通知媒体服务器需要等待的时间 $T\_wait - L\_head\_seg / V\_bit\_rate$，等待结束后转5；

⑤从硬盘读取文件第一段，并发送给媒体服务器，直到第一段结束；

⑥请求磁带库，要求从磁带读取文件第二段(按照前面的条件约束，此处可以立即返回)；

⑦从磁带读取文件第二段，并发送给媒体服务器，直到第二段结束。

4 仿真试验结果及性能分析

我们模拟了在采用这种数据放置策略，且设定 $T\_queue=200s$ 的 HSM 系统中，用户请求的执行情况。系统设定同0节，得到的实际等待时间分布曲线如下。

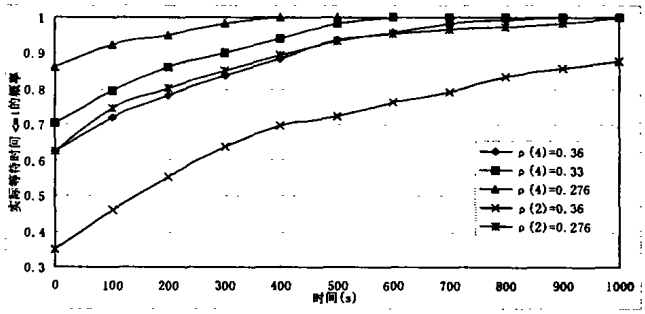

图5 访问 HSM 系统的用户请求等待时间分布曲线

可以看出，在负载不是很重的情况下，大部分请求都可以得到即时响应。此时，由公式4，媒体对象第一个段的平均长度：

$$E(L\_head\_seg)=0.5*(12.8+51+\frac{21000}{300}+200)$$
$$=166.9(\text{Mbytes})$$

由于媒体对象的大小为2Gbytes，因此预置在磁盘的头部数据占其中的166.9/2000=0.084。

因此当 HSM 系统的总容量为 R 时，所需要的磁盘空间约为 R * 0.084。假设媒体服务系统一共拥有2000个媒体对象，则 HSM 系统总容量为4T，而 HSM 系统中需要的磁盘空间约为335G。

(下转第28页)

络中维护两种截然不同的路由协议，也很少使用。a)项需要在同一台路由器中运行两个互相独立的进程，这种机制称为"ships in the night"。而d)项只需要在一台路由器中运行一个进程。显然，a)项需要占用路由器更多资源，但由于a)项可以生成两个独立的IPv4和IPv6的路由表，IPv4和IPv6可以走不同的路由，从而在逻辑上可以形成不同的IPv4的网络拓扑和IPv6的网络拓扑。d)项占用路由器较少的计算资源，但由于IS-ISv6中IPv4和IPv6的拓扑计算是在同一个SPF计算进程中完成的，IPv4和IPv6的流量需要遵循同一个拓扑，在灵活性上不如a)项，同时在试验阶段，如果采用d)项，IPv6路由协议的崩溃将会导致IPv4的同时崩溃，而采用a)项则没有这种问题。如果IPv4的流量和IPv6的流量遵循同一个逻辑拓扑结构，此时采用d)项就会比较方便，如果采用a)项，在网络拓扑修改时，需要同时修改两个协议，容易造成拓扑的不一致，虽然目前有的厂商推出或正在研究IS-IS多拓扑计算，但还没有在国内外的IPv6网络上得到可靠性验证。

3.2.3 *网内路由协议的具体选择* 可使用IBGP和IS-IS链路状态协议，同时承载IPv4和IPv6，骨干网采用IBGP承载用户路由信息，IS-IS承载互联链路路由信息的路由政策。一是IS-IS路由协议网络可扩展性强，利于以后的扩容，IS-IS具有很好的分层、分域能力，适用于大型网络；二是IS-IS路由协议可只承载网络链路互联信息，不承担用户信息，路由表可以大大地简化和相对稳定。因此，网络链路稳定的情况下，在上百台路由设备的大型网络中应用，也可以满足要求。

对于用户路由信息的承载，选择IBGP可以提高全网路由的稳定性和层次化。因为BGP使用TCP会话保持对端之间的连接，所以具有面向连接的可靠性。另外，由于采用应用层数据报，因此，可以将于大量的路由刷新在一次会话中完成，节约了网络带宽。另外，由于用户路由与链路路由的分离，提高了网络的稳定性和层次化，便于网络维护。但由于BGP路由并不直接反映用户链路状态变化，因此，需要使用路由注入或手工配置的方式将用户可达信息在BGP路由表中传递，这会带来手工操作的工作量。另外，因为用户路由信息与网间互联信息都在BGP中承载，因此需要制订严格的维护规范，避免非法路由广播影响网络的稳定性。为了保证骨干网内路由条数尽量少，减少维护量，建议对地址进行规划，并且边缘路由器在广播IBGP路由时尽量广播完整的地址块。

小结 本文讨论了IPv6网络核心路由协议、选择原则及具体的选择。需注意，一般不推荐核心骨干网在隧道工作方式下运行OSPFv3或IS-ISv6，因为不论是OSPF还是ISIS不像BGP是通过TCP会话交换路由更新，在IPv4网络环境恶化的情况下，会引起SPF的频繁计算，造成IGP路由的不稳定。另外，实际的网络建设中，还需考虑网间国际出口BGP路由策略、网间国内出口BGP路由策略、网内IBGP路由策略、网内IGP路由策略、用户路由策略等，需对核心路由协议给予足够的重视。

致谢 感谢中国联合通信有限公司技术部技术开发处王明会博士、杨征先生与作者有益的讨论。

参考文献

1 Deering S, Hinden R. Internet Protocol, Version 6 (IPv6) Specification. RFC2460, 1998
2 Huitema C. 新因特网协议IPv6. 陶文星等译. 北京：清华大学出版社，1999
3 RFC2185. Routing Aspects of IPv6 Transition. 1997
4 RFC 2080. RIPng for IPv6, 1997
5 RFC2740. OSPF for IPv6, 1999
6 draft-ietf-isis-ipv6-02.txt, Routing IPv6 with IS-IS, 1999
7 Bates T, Chandra R, Katz D, Rekhter Y. Multiprotocol Extensions for BGP-4. RFC 2283, Feb. 1998

（上接第25页）

假设VFS的磁盘中存储200个媒体对象，则根据zipf分布的规律，用户请求的媒体对象位于VFS上的概率为$\sum_{n=1}^{200}\frac{C}{n}=0.7183$，（其中$C=1/\sum_{i=1}^{2000}\frac{1}{i}$）。采用普通HSM系统时，有28.17%用户请求需要等待。

采用基于预置的数据放置策略后，在每小时100个用户请求的负载下（对应于$\rho(4)=0.33$），访问HSM的请求中有70.5%可以即时响应。因此，需要等待的用户点播请求占$0.2813*(1-0.705)=0.083$。

若以增加VFS磁盘空间的方式使同样91.7%的请求得到即时响应，则由于$\sum_{n=1}^{1015}\frac{C}{n}=0.917$，（其中$C=1/\sum_{i=1}^{2000}\frac{1}{i}$），VFS需要存放1015个媒体对象，相应的需要增加的磁盘空间为$(1015-200)*2G=1.6T$。

总结 本文通过剖析三级存储设备的访问特性和媒体服务系统的特点，提出HSM系统中面向流媒体服务的一种数据放置策略。分析和仿真试验表明，采用这种策略，只需付出较小的存储代价，就可以使得HSM系统具有较高的即时响应性能，从而大大提高整个媒体服务系统的性能。

参考文献

1 Barnett S A, Anido G J. A Cost Comparison of Distributed and Centralized Approaches to Video-on-Demand. IEEE Journal on Selected Areas in Communications, 1996, 14(6): 1173～1183
2 Brubeck D W, Rowe L A. Hierarchical storage management in a distributed VOD system. IEEE Multimedia, 1996, 3(3): 37～47
3 Christodoulakis S, Triantafillou P, Zioga F. Principles of optimally placing data in tertiary storage libraries. In: Proc. of Intl. Conf. on Very Large Data Bases, Athens, Greece, 1997. 236～245
4 Li J, Prabhakar S. Data Placement for Tertiary Storage. In: the 19th IEEE Symposium on Mass Storage Systems, Maryland, 2002
5 Ghandeharizadeh S, Dashti A, Shahabi C. A pipelining mechanism to minimize the latency time in hierarchical multimedia storage managers. Computer communications, 1995, 18(3): 170～184
6 Prabhakar S, Chari R. Minimizing Latency and Jitter for Large Scale Multimedia Repositories through Prefix Caching. International Journal on Image and Graphics (IJIG), 2003, 3(1): 95～117
7 Johnson T, Miller E L. Performance measurements of tertiary storage devices. In: Proc. of 24th Intl. Conf. on Very Large Data Bases, New York, 1998. 50～61
8 Yang Daoliang, Ren Xiaoxia, Chang Ming. Study on Data Replacement Algorithm in Continuous Media Server with Hierarchical Storage. In: 16th IFIP World Computer Congress, Beijing, Aug. 2000
9 Fishwick P A. Simpack: Getting Started with Simulation Programming in C and C++. In: 1992 Winter Simulation Conf. Arlington, VA, 1992. 154～162
10 SimPack Toolkit. http://www.cise.ufl.edu/～fishwick/simpack/simpack.html