

WebOS: Internet 操作系统^{*}

武秀川^{1,2} 鞠九滨²

(烟台大学计算机科学系 烟台 264005)¹(吉林大学计算机科学系 长春 130023)²

Abstract The development of Internet, protocols and standards such as S-HTTP, HTML, CGI, MIME, PERL has made World Wide Web been greatly used intensively. It will support the distributed computing based-on the Web as well as multiple-purpose and service-oriented. WebOS is an operating system based-on the Web, providing basic operating system services needed to build applications that are geographically distributed, highly available, incrementally scalable and dynamically reconfigurable. In this survey of WebOS, components of WebOS, their relationship between each other are described. At the end of this paper, the authors propose some problems about WebOS

Keywords WWW, WebOS, WebFS, Smart-Client, Rent-A-Server, Resource Locating, Security

1 引言

WebOS 概念最早由 Syracuse 大学 NPAC 的 G C Fox 等人于 1995 年提出,当时称之为 WebWindows。他们认为 WebWindows 将作为一个开放的、公用的和模块化的 Internet 分布式操作系统。文[11]介绍了他们在此概念的基础上利用许多已有的 Web 技术所建立的一个并行程序设计环境和所提供的虚拟程序设计实验室 VPL。Berkeley 大学的 WebOS 工作组提出了 WebOS 的完整概念,并作了卓有成效的工作,实际上 WebOS 基于 WWW 技术建立了一个计算网格^[16]。文[4]对 WebOS 的各基本组成部分做了概括性的介绍。本文基于 WebOS 的概念对其组成部分作了分析,着重从操作系统资源调度的角度出发,揭示各组成部分之间的相互联系,指出了某些算法中的不足,为今后的研究提出了一些思路。

WWW 是 Internet 最具活力的应用之一,其基本结构是用 URL 命名资源、用 MIME 绑定资源、用 HTTP 传输资源、用 HTML 显示资源、用 CGI 为 HTTP 服务器提供基本的延伸性。此外,引入的许多 Web 新技术如 Client Pull、Server Push、HTTP Cookie、Java Script 和 Java Applet、XML、WinCGI 及 ISAPI 应用程序 ISAPI 过滤器、Active X、HOTJava 浏览器和视图等大大扩展了 Web 的功能。但仅对 Internet 地理上分布的只读应用提供了方便,而对地理上分布的计算资源如远程 CPU、内存和文件的访问仍是极其困难的,程序设计者必须以一种特定的方式进行程序设

计,造成了人力和计算机资源的极大浪费。WebOS 提供一种全新的 Internet 服务范例,旨在为地理上分布的、高效的、可逐步扩充的、可动态重构的 Internet 广域分布式应用提供操作系统服务,进而使用户无缝地、透明地访问 Internet 上任意站点的任何资源,并且可把某种服务动态地复制和迁移到其它的 Internet 资源上,甚至全部迁移到客户机上去。WebOS 的这种动态可重构和地理上可移动的服务具有明显的优越性^[4]: 1)运行在客户端特定的服务方式屏蔽掉了 Internet 或服务器的失效,从而提高了端对端的有效性;2)通过动态地把信息移动到尽可能地靠近提出请求的客户机,则能减少网络延迟、拥塞和开销,提高性能价格比;3)通过动态补充资源以处理峰值请求,提高了处理突发事件的能力。

2 WebOS 的组成

为了实现上述目标,WebOS 应解决资源分配调度过程中的全局资源定位、广域文件使用、安全和身份鉴别、认证以及远程进程控制等问题,它们综合提供对广域分布式应用程序的操作系统服务。

2.1 全局资源定位

广域应用服务是在地理上分布的。为了解决随着 WWW 快速增长而导致的响应慢、网络拥塞及“热点”等问题,客户机应用程序应能动态地确定 Internet 中提供优质服务的服务器,进而使用户获得可扩充的 Internet 服务。为此必须解决负载均衡、容错等问题。

注意到网络延迟以及带宽限制正在成为网络服务

^{*} 国家自然科学基金资助项目(69873018)。武秀川 副教授,博士研究生,研究领域为分布式系统,鞠九滨 教授,博士生导师,研究领域为计算机网络、分布式系统。

的瓶颈,客户机有可能处于空闲状态。如果能把服务器的某些功能以及状态迁移到这些空闲的客户机上可能改进总的服务质量。一些 Internet 技术的出现为这种迁移提供了可能性。首先,通过 Web 浏览器,用户可方便地对最普遍的 Internet 服务如 HTTP、FTP 以及搜索引擎等进行访问;其次,与机器无关的 Java 语言可

非常容易地把功能迁移代码分布到多个不同的平台上,这种“智能客户机”^[1,2]与传统的“瘦客户机”完全不同,它对最终用户提供负载平衡和容错功能。

两个相互协作的子程序(Applet)组成了智能客户机,见图 1。

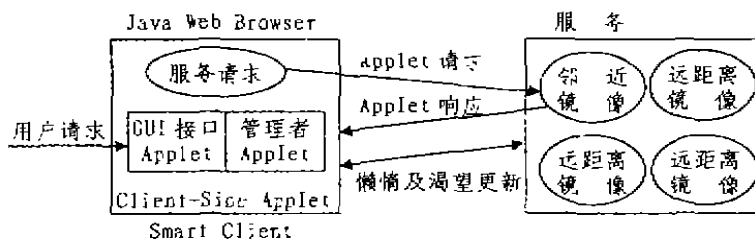


图 1 Smart Client 体系结构

图形用户接口(GUI)Applet 提供服务接口并把用户请求传递给管理者 Applet。对每个请求,管理者 Applet 根据负载、有效性等信息选择一个可提供最佳服务的服务器,以满足客户需求。该方式中,由智能客户机实现的服务可由若干数量的同级别服务器镜像实现。负载平衡及容错算法的选择有两种^[3-9]。1)随机的服务器选择,该算法实现容易且不要求服务器更新,但负载平衡及满足客户要求性能较差;2)维持指定服务的服务器配置文件算法,即 Smart Client 根据服务器的硬件特性和负载以及网络连接性等信息以及客户机的位置估计每一个服务器的可用带宽而选择服务器,该算法可获得很好的负载平衡性能,但根据什么原则估计带宽目前尚无确定的算法。

上述负载平衡算法,结合服务器是否活跃或超时等决定服务器是否失效,一旦失效,Smart Client 再调用负载平衡算法选择另一个服务器。为了做出负载平衡决策,客户机需要一个最新的服务器配置文件,可用懒惰或渴望技术更改管理者服务状态的变化,并且通过在性能及语义之间做出权衡,很好地维持了一致性。在 Web 这样的环境下,懒惰技术可以减少网络流量,如当服务器回答客户请求时可用“附载”(如可作为 HTTP 头的一部分)把更新信息一同送到客户机。

Smart Client 从逻辑上把指定的服务器功能迁移到客户机上,从而动态地实现负载平衡和透明的容错功能。这种透明性包括不必对服务器的拓扑结构作任何限制,可以根据指定的服务方式选择指定的服务器。同时为了有效地减少用户每次请求导致的延迟,指定服务的 Java Code 使用自举 Applet 机制仅在第一次访问服务时传送 Applet,并使用 Jbox 机制将其缓存到磁盘,这样在以后各次访问中不必再传送 Java Code。

显然 Smart Client 较其它方案更能有效地提高整体服务性能,更加具有智能化。HTTP redirect^[3]机制通过改向使客户机请求转到另一台服务器从而实现负载平衡,但除了对客户机请求的双倍延迟外,单点失效问题仍然存在。Magic Router^[4]通过把一个逻辑 IP 地址映射为多个服务器,并且在把 IP 包送到目的服务器之前对其进行观察和可能的修改,进而实现负载平衡和容错,但缺陷是它要求一个特殊的网络拓扑结构。Fail-Safe TCP^[5]用两台独立的机器通过复制 TCP 状态实现容错,但除要求多余的硬件资源之外,也不能解决前端机变为瓶颈的问题。Active Network^[6]允许包在传输到目的服务器的过程中在智能化的网络路由器里进行计算,其主要目的是能够更容易地扩展网络协议进而提供新的服务,它也可能提供负载平衡及容错。Smart Client 可看作是 Active Network 的逻辑扩充。Active Names^[7]系统通过在可编程路由器内包一级把服务名字映射到一组地址中去,然后动态地把包传送到目的服务器。Active Names 综合了 Active Network 和 Active Service^[8]。当包流经路由器时 Active Networks 提供任意的计算,完全透明地使包到达目的地,但是代价太大,主要表现在表示端对端语义的极大困难性上。文[2]考虑到 Internet 路由体系结构的不同,通过构造一组经由路径质量测量形成的另外的路径(而不是系统设定的路径)组成一个新的综合路径,这种综合路径总是从连接 Internet 任意两点的多个路径中选取最佳路径。

Smart Client 的局限性是目前只可以迁移管理者 Applet,不能从服务器迁移其它代码,一定程度上限制了它的作用。

2.2 广域文件系统 WebFS

Internet 中存在着各种各样的分布式应用,它们

地理位置分散, 大小规模不同, 性质各异。目前所采用的网络通信传送信息和远程服务方法明显地增加了网络通信量和服务器负载, 使性能下降。采用缓存方法可使大量的远程访问由本地缓存器有效地处理, 以节省在网络上的传送时间, 尤其当文件访问流表现为访问的局部化时性能最好^[4,5], 但缓存的一致性问造成了极大的困难, Web 中的某些不经常写的访问模式如 Internet 新闻服务只需弱一致性, 这可以大大地减少系统开销或确保网络失效不会延迟更新, 但在经常写的访问模式应用中则需很强的一致性。解决一致性问

题的机制在性能、网络通信量和服务器负载方面产生重大开销, 必须很好地解决共享语义问题^[1,2]。WebFS^[6]实现了对全局名字空间一致性的缓存访问, 并作为一个完全意义上的文件系统接口支持任意的文件和目录操作。WebFS 使用基于 URL 的名字空间, 其守护进程使用 HTTP 对标准的 Web 站点进行访问。考虑到可移植性, WebFS 建立在 UNIX V 结点层, 并与虚拟内存系统紧密结合起来。WebFS 系统结构见图 2^[6]。

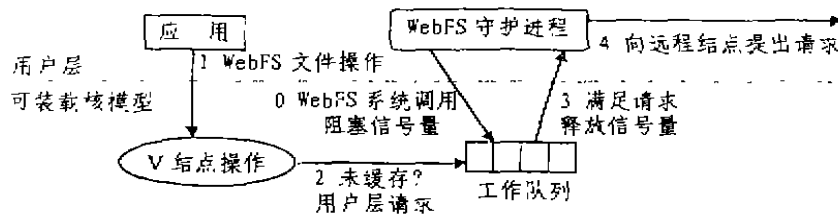


图 2 WebFS 体系结构

该结构的优点是^[4,6,8]: 1) V 结点层的接口允许应用程序访问全局 HTTP 名字空间; 2) 与虚拟内存系统紧密结合, 因此提供高性能的缓存访问; 3) 大多数的 UNIX 变种以及 Windows NT 支持 V 结点接口, 可简化移植到新平台的入口; 4) 许多文件系统函数和数据结构维持在用户层, 便于实验和简化调试及用户开发。

缓存策略的选择至关重要。对于一个文件被众多用户共享读而仅有一个用户在任意时刻对其写、修改的应用, 采用“最后写者获胜”缓存一致性策略。对于 Internet Chat 等应用, WebFS 提供“只增加”一致性缓存策略。Internet Chat Room 实际上就是 WebFS 文件, 其读操作是幂等的; 而写操作是原子的, 即对于一个写操作, WebFS 把它的所有客户机进行更新, 并以一种懒惰方式传播到其它 WebFS 服务器中。这种一致性缓存策略还可用于其它分布式应用如共享白板。大量的“Internet Push”应用服务如视频会议、新闻服务等, 客户机关心所有的更新, 对于这种被广泛共享和频繁地更新数据文件的应用, WebFS 采用基于 IP 广播的一致性策略, 使用路由器支持, 仅沿着必要的路由广播复制包, 同时还可以简化服务器设计和发现错误。采用广播一致性协议涉及到用于广播的 IP 地址的数目有限、路由表庞大和可靠性等几个问题^[7,8], 可以分别采用 IPv6、压缩技术以及 SRM 方法解决。

与 WebFS 相比, AFS 的致命弱点是不支持 HTTP 名字空间; UFO^[11]只对 FTP 提供松散的一致性保证, 而对 HTTP 却不提供, 另外通过 FTP 访问远程文件需要一个帐号, 限制了 UFO 系统的广泛使用;

Alex^[9]较之 UFO 增加了一致性保证, 但其只读访问的一致性策略并不适宜于大多数的 Internet 分布式应用。

2.3 安全和认证系统

WebOS 允许用户无缝地访问 Internet 任一网站上的任何文件、数据及计算资源, 但这一访问易受各种潜在的错综复杂因素的影响, 必须建立一个完善的广域安全系统。公钥密码学可以提供广域应用中的安全性, 它消除了与一个信任的第三方的同时通信的需要。当用户程序访问一个远程资源时, 为使其完成任务, 应赋予该任务一组最小的特权。这样当远程资源出现故障时可使该任务的数据、身份证以及私人文件得到最好的保护。为此, 广域安全系统必须提供不同管理域主机之间权限的细粒度转让。Kerberos 系统虽然解决了转让问题, 但仅局限于局域网, 若将其扩充到广域网, 则面临着较大的网络延迟和因没有冗余而使单点危害可能危及整个系统等问题。CRISIS^[6]实现了广域安全认证和访问控制, CRISIS 并不是针对任何已知的安全攻击而设计的, 它使用众多已有协议如 SSL、S-HTTP、PEM 以及 X.509 等。此外通过对冗余、缓存、轻量灵活性、局部化操作等原理的使用, 来增加广域分布系统的安全性, 同时为资源提供者提供认证请求确认和授权访问其资源的机制, 最终达到用户安全地访问 Internet 全局资源。

CRISIS 采用两种类型的证书^[6,11-13]: 身份证及转让证书。身份证由 CA 签署, 并把公用密钥映射到主机。为增加安全可靠, CRISIS 还确定一个通信双方

局部信任的代理 OLA,对证书进行联名签署,主要用于检查一个主机所做的陈述,特权声明或特权的转让是否已被撤消或对刷新证书提供担保。CA 签署的特权通常具有较长时延(数周),而 OLA 签署的特权具有较短时延(数小时)。这种隔裂 CA/OLA 的方法提供明显的优越性^[4]:1)只有同时破坏了 CA 与 OLA 才可以成功地窃取密钥;2)由于 CA 所签署的证书具有较长的时延,即 CA 是离线的,攻击一个离线实体是很困难的,因此增加了安全性;3)恶意的 CA 如果不与 OLA 串通勾结,则不可能废除用户的密钥而颁布一个新的身份证书以及假扮用户;4)一个恶意的 OLA 可以拒绝承认进行了攻击,CA 仍然可以利用不同的 OLA 重新颁布一个新的证书;5)由于 OLA 所签证书具有一段较短的时延,所以双方通信时不需要与第三方的同时通信,提高了系统性能。当发生某种意外(如密钥被窃取)或一些通常的操作(如用户对话结束退出登录或用户决定取消作业)时则有效地撤消一个特定的权限。另外当证书期满其相应的特权也被撤消,OLA 不再认可该证书。

CRISIS 的最大特色是引入了转让证书的概念,即把一个主机特权的子集转让到另一个主机。转让证书过程可以递归进行从而形成一条链,链上的每一个后续链接主机所获得的转让的权限只能缩小而不能增加。用此方法提供不同管理域之间权限的细粒度转让。

目前,CRISIS 采用“Push”技术提供授权时的信任,从而降低了系统复杂度和提高了性能,但相对于“Pull”技术,灵活性较差。

2.4 安全远程进程控制

虽然使用 Web 浏览器连接到一个远程 Web 服务器的最主要目的是获得对信息的访问,但信息并不是唯一可用的资源。通过 Web 系统,用户也可以访问远程高性能硬件及计算资源^[17]。WebOS 的远程进程控制系统可确保经认证的用户进程在远程机上安全地运行,并且与在本地运行同样简单。为了维护本地系统的完整性并且确保运行的各进程不会相互干扰,每个 WebOS 机的资源管理者为每个运行的进程形成一个虚拟机。这些虚拟机与 CRISIS 安全系统相互配合,使得在虚拟机中运行的进程可对本地资源进行不同的访问。

用不同方法如 SRHA 和 Janus 等提供的虚拟机使用操作系统跟踪能力截获可能危害系统完整性的系统调用。如果一个应用程序试图进行一个具有潜在危害性的操作时,则拒绝系统调用。由于通常情况下目标系统调用很少,所以这一过程总的代价开销可以忽略不计。形成虚拟机只为在远程结点运行进程提供条件。远程运行进程包括进程转移和进程迁移。进程转移虽然

实现简单,但会有很大的时间延迟。在虚拟机环境中实现有通信关系的进程迁移,要解决如下两个关键问题:Socket 通信的原子性,即进程迁移应选择 Socket 在不进行通信的时刻做检查点;相关进程通信的处理,即进程迁移时应保证被迁移进程与系统中的多个其他进程的通信不受进程迁移的影响。在解决虚拟机的进程通信机制中,既要使进程通信成为系统的瓶颈,又要使整个系统的开销增大。如果在 WebOS 的系统层实现进程迁移,这将是一个很有意义的工作,因为它可以对用户实现迁移透明性,但系统层实现的进程迁移依赖性很强,会造成移植的困难。

2.5 Rent-A-Server: WebOS 的一个负载平衡方法

目前所采取的分布 HTTP 服务器负载的方式主要有以下几种^[3]:1)HTTP 服务器通过一组固定的服务器复制其数据,然后随机地把负载分布到这一组服务器上,以此来满足峰值要求;2)使用镜像网站,这种方式要求各镜像维持数据一致性;3)缓存代理主要提供防火墙安全,还可以减轻服务器负载和局部提高网络性能,但它仅能缓存数据页,并且服务器把代理视为普通的客户机,因此代理提供给客户机的可能是已过时的数据。这些方法或者由于用户过多的参与或者受到 HTTP 缓存一致性协议的限制,使得响应较差,或者要求每个站点购买足够的计算资源和网络带宽满足峰值要求。用 Rent-A-Server 方法,一个超载的 HTTP 服务器能够把负载卸载到空闲的、离请求服务的客户机距离最近的服务器上去,而该服务器使用 WebFS 系统缓存(一致性的)来自于超载服务器上的数据。Rent-A-Server 通过透明的、自动的、动态的补充资源的方式满足广域系统的峰值要求。Rent-A-Server 结构见图 3^[4]。

其中的智能客户机用来对 HTTP 服务实现负载平衡访问,WebFS 提供透明的 URL 应用访问以及缓存一致性,以避免对过时的数据的访问。HTTP 服务器周期性地把负载信息传送到处于中心位置的负载守护进程中。为了响应更新了的负载信息,负载守护进程把所有服务器的状态信息传送给每一个服务器。HTTP 服务器把这些状态信息作为 HTTP 响应头部传送到智能客户机中。智能客户机把这些状态信息作为下次请求对服务器的选择依据。负载守护进程根据目前的负载信息和客户机的访问模式决定“除掉”还是“租一个服务器”(生成另一个服务器副本)。如果需要“租一个服务器”,则该服务器的位置应是在考虑服务器访问模式的情况下离客户机最近的,以保留带宽和使客户机延迟最小。选择了目标服务器后,负载守护进程与 Rent-A-Server 的资源管理者进程建立一个 SSL

- ented Programming Systems, Languages and Applications in 1992, Oct. 1992
 - 12 Horne M, et al. Designing Meta-Interfaces for Object-Oriented Operating Systems. In: Proc. of 1997 IEEE Pacific Rim Conf. on Communications, Computers, and Signal Processing, 1997 989~992
 - 13 Hildebrand D. An Architectural Overview of QNX. In: Proc. of the Usenix Workshop on Micro-kernels and Other Kernel Architectures, 1992. 113~126
 - 14 侯业勤, 张菁编译. 分布式嵌入式实时操作系统 QNX 宇航出版社, 1999
 - 15 Bowman I. Conceptual Architecture of the Linux Kernel. Jan. 1998 (Available from <http://plg.uwaterloo.ca/~itbowman/CS746G/al/>)
 - 16 陈莉君编著. Linux 操作系统内核分析. 人民邮电出版社, 2000
 - 17 Bach M J. The Design of the UNIX Operating Systems. Prentice Hall, Englewood Cliffs, NJ, 1986 (见: 陈葆钰等译. UNIX 操作系统设计. 北京大学出版社, 1989)
 - 18 Acetta M, et al. Mach: A New Kernel Foundation for UNIX Development. In: Proc. of the Summer 1986 USENIX Conf., 1986. 93~112
 - 19 Black D L, et al. Microkernel Operating System Architecture and Mach. In: Proc. of the USENIX Workshop on Microkernels and Other Kernel Architectures. Seattle, Washington, Apr. 1992 11~30
 - 20 Guillemonet M. The Chorus Distributed Operating System: Design and Implementation. In: ACM International Symposium on Local Computer Networks, Firenze, Apr. 1982. 207~223
 - 21 Rozier M, et al. CHORUS Distributed Operating System. Computing Systems Journal, The USENIX Association, Dec. 1988, 1(4). 350~370
 - 22 Probert D, Bruno J. Building Fundamentally Extensible Application-Specific Operating Systems in SPACE. [Technical Report TRCS95-06]. Computer Science Department, UC Santa Barbara, Mar. 1995
 - 23 Veitch A C, Hutchinson N C. Kea-A Dynamically Extensible and Configurable Operating systems Kernel. In: Proc. of the Third Conf. on Configurable Distributed Systems, 1996
 - 24 Engler D R. The Design and Implementation of a Prototype Exokernel Operating System. [MA Thesis] MIT, Oct. 1998
 - 25 Deitel H M, Kogan M S. The Design of OS/2. Addison Wesley, NY, 1992
 - 26 Marsh B D. Multi-Model Parallel Programming. [PhD Thesis]. Department of Computer Science, University of Rochester, Jan. 1992
 - 27 Assenmacher H, et al. The PANDA System Architecture--A Pico-Kernel Approach. In: Proc. of the Forth Workshop on Future Trends of Distributed Computing Systems, Sep. 1993. 470~476
 - 28 Murray K, et al. Design and Implementation of an Object-orientated 64-bit Single Address Space Microkernel. In: Proc. of the 2nd USENIX Symposium on Microkernels and Other Kernel Architectures, Sep. 1993. 31~43
 - 29 Heiser G, et al. Mungt: A Distributed Single Address Space Operating System. [Technical Report UNSW-CSE-TR-9314]. School of Computer Science and Engineering, University of New South Wales, Sydney, Australia, Nov. 1993
 - 30 Chase J S, et al. Sharing and Protection in a Single Address Space Operating System. ACM Transaction on Computer Systems, 1994(12): 271~307
 - 31 Roscoe T. The Structure of a Multi-Service Operating System. [PhD Thesis]. Queens' College, University of Cambridge, Apr. 1995
 - 32 Wilkinson T, et al. Single Address Space Operating Systems. [Technical Report UNSW-CSE-TR-9504]. School of Computer Science and Engineering, University of New South Wales, Sydney, Australia, Nov. 1995
 - 33 张载鸿编. 局域网操作系统 DOS 高级技术分析. 国防工业出版社, 1988
 - 34 INTEL 资料. Intel Architecture Software Developer's Manual. 1997
 - 35 孟庆余. 系统软件平台 COSA 的设计与实现. 电子学报, 1999, 27(2)
-
- (上接第 5 页)
- 11 Dincer K, Fox G C. Design Issues in Building Web-Based Parallel Programming Environments. In: Proc. of the Sixth IEEE Intl. Symposium High Performance Distributed Computing. Portland State University, Portland, Oregon, Aug. 1997
 - 12 Dincer K, Fox G C. Using Java and JavaScript in the Virtual Programming Laboratory: A Web-Based Parallel Programming Environment. Concurrency: Practice and Experience, June, 1997
 - 13 Fox G C, Furmanski W. The Use of The National Information Infrastructure and High Performance Computers in Industry. [NPAC Technical Report SCCS-732]. July, 1995
 - 14 Alexandrov A D, et al. Ufo: Personal Global File System Based on User-Level Extensions to the Operating System. In: Proc. of the 1997 USENIX Technical Conf. Anaheim, CA, Jan. 1997
 - 15 Gadde S, Chase J, Rabinovich M. Web Caching and Content Distribution: A View From the Interior. In: Proc. of The 5th International Web Caching and Content Delivery Workshop, May 2000
 - 16 Anderson E, Patterson D, Brewer E. The Magicrouter, an Application of Fast Packet Interposing. <http://HTTP.CS.Berkeley.EDU/~eanders/magicrouter/>
 - 17 Tennenhouse D, Wetherall D. Towards an Active Network Architecture. In ACM SIGCOMM Computer Communication Review, April, 1996. 5~18
 - 18 Dongarra J, et al. High Performance Computing Today. To Appear in FOMMS 2000: Foundations of Molecular Modeling and Simulation Conference
 - 19 Yu Haifeng, Vahdat A. Design and Evaluation of a Continuous Consistency Model for Replicated Services. In: Proc. of the Fourth Symposium on Operating Systems Design and Implementation (OSDI), Oct. 2000