Google Linux Cluster 的系统结构分析

余一娇[1,2]
（[1]华中师范大学语言学系 武汉 430079）
（[2]华中科技大学计算机学院 武汉 430074）
E-mail: yjyu@mail.ccnu.edu.cn

摘 要：Google 是当前最有影响的 Web 搜索引擎，它利用一万多台廉价 PC 机构造了一个高性能、超大存储容量、稳定、实用的巨型 Linux 集群。本文是从计算机系统结构的角度分析 Google 集群系统的逻辑和物理构造方法、可靠性、可扩展性、可用性、并行性。文中重点介绍了 Google 集群的逻辑结构和物理结构、分布式文件系统和超大容量存储器的实现方法。文中分析认为 Google 集群针对 Web 搜索需求的特征，用低成本实现了高可用、高性能集群的方法是并行机设计、开发一个成功典范,这种严格追求性价比的设计方法值得借鉴。
关键词：Google 集群 系统结构 文件系统 可靠性 可扩展性 并行性

The System Architecture Analysis of Google Linux Cluster

Yijiao Yu[1,2]
[1] Department of Linguistics, Central China Normal University, Wuhan, 400079
[2] College of Computer Science and Technology,
Huazhong University of Science and Technology, Wuhan, 430074
E-mail: yjyu@mail.ccnu.edu.cn

Abstract As the most popular and successful web search engine and a super Linux cluster, Google integrates more than 15,000 commodity personal computers (PC) to support the information storing and processing. The outstanding features of Google Linux Cluster are high performance-cost ratio, high availability and excellent scalability. We investigate it from the view of computer system architecture in this paper. The logical architecture and physical structure, distributed file system, reliability, availability, and huge storage systems are discussed respectively, and the related technologies are analyzed and reviewed. Finally, in the light of Google, we conclude that we should build a high-available cluster with the as least cost as possible, and select the appropriate technologies to satisfy the real application requirements.
Keywords: Google; Cluster; System Architecture; File System, Reliability; Scalability, Parallelism

1. 前言

Google 公司于 1998 年由 Stanford 大学计算机系的两个博士研究生 Sergey Brin 和 Larry Page 创立[1]；2000 年被《Internet Life》杂志评为“互联网上最好的搜索引擎”[1]，今年被世界品牌实验室评为 2005 年世界品牌 500 强中的第三名[2]。与曾经最有影响的搜索引擎 Yahoo 相比，Google 在创立之初在技术和经济资源方面并不具备太多优势，是什么原因导致网络用户放弃 Yahoo、接受 Google,我认为对搜索结果质量评价的正确观点可能是最主要的原因。1998 年期间，网络中的 Web 页面数量较小，可以采用手工方法对每一个页面进行人工分类。对一个用户查询请求，可以查全所有的 Web 页面，然后反馈所有相关页面链接。然而在 1998 年，Sergey Brin 和 Larry Page 在《Computer Networks》杂志上合作发表论文[3]却提出了不同的观点，他们指出：Google 追求的高质量网络搜索不是拥有尽可能高的查全率，而是反馈给用户的前几十条链接必须具有很高的精确率。基于该理念，Sergey Brin 和 Larry Page 在 Google 搜索引擎中采用了根据链接数量来判定页面质量的 PageRank 算法，该算法后来成为网页评价研究中的一个很著名的算法。由于 PageRank 算法的分析和

改进工作已有很多相关研究和评论，本文不对此进行讨论。本文只关注 Google 服务器技术。

1998 年 Google 创立之初，该公司的数据中心放置在 Larry 在 Stanford 大学的宿舍内，且公司的第一笔资金只有 100 万美元[1]。从当时的服务器、大型机价格来看，100 万美元即使全部用来购买服务器，也不可能购买十分先进、高档的服务器或大型机。数据中心放置在学生宿舍，受房间面积限制，大型机及附属设备所需占地面积也不可能得到满足。同时大型机对降温要求很高，一般学生宿舍也很难得到保障。关于大型机所需供电、降温需求，以及普通房屋供电负荷相关参数可参考文献[4]。因此根据以上信息，可判断 1998 年 Google 不太可能采用、也很难有条件采用传统的大型机作为搜索引擎的服务器。

1998 年以后互联网发展速度越来越快，无论是 Web 页面数量，还是用户提交的搜索请求数量都在短短几年内增长了数十倍甚至上百倍。Google 搜索引擎却没有因为计算量和存储量高速增长而变得不堪重负，相反它一直争取在 0.5 秒时间内处理完成用户请求。如今 Google 完成查询请求后，还把查询所耗费的时间反馈给用户。以 2005 年 4 月 21 日上午 10 时 51 分在华中科技大学校园网内输入“华中科技大学”一词进行检索为例，Google 反馈了 917,000 个查询结果，查询时间却仅为 0.22 秒。对同一个检索词在不同时间段进行检索，搜索引擎消耗的时间不尽相同，因为处理一个请求所需时间不仅受搜索引擎中待检索页面的数量、大小限制，也与整个系统当前的繁忙程度有关。在 0.5 秒内完成数据检索，Google 服务器应该是高性能计算的一个成功典范。作为面向全球用户的商业搜索引擎，由于全球存在时差，Google 必须保证每天 24 小时都能正常运行，这对 Google 服务器的可用性提出了巨大的挑战。

Google 本是数学中的一个名词，它表示一个十分巨大的数：1 后面跟 100 个 0（即 10^{100}）[3]。Sergey Brin 和 Larry Page 使用 Google 作为自己的搜索引擎和公司名的主要原因是希望自己设计的 Web 搜索引擎将来能够支持十分巨大的 Web 页面检索[3]。至 2005 年 4 月 21 为止，Google 中所收集的 Web 页面数量已经达到 8,058,044,651 张（见 Google 页面）。虽然该值与 Google 所描述数量还相差甚远，但它成为如今世界上收集 Web 页面最多的搜索引擎。

本文内容结构安排如下：第二部分介绍 Google 集群（机群）的逻辑结构和物理结构；第三部分讨论 Google 的文件系统；第四部分介绍 Google 的存储系统和存储策略；第五部分讨论 Google 的高可用性和可靠性；第六部分讨论 Google 集群在计算和存储中的高并行性。最后根据 Google 集群的成功经验对大型机或大型集群开发提出一些思考。

2、Google 集群的逻辑结构和物理结构

为 Google 搜索引擎提供硬件支持的不是传统的大型机和服务器，而是技术含量低、廉价的集群技术。至 2003 年 4 月，Google 集群已集成 15,000 台 PC 机，成为当时世界上最大的 PC 机集群系统[4]。文献[5]中指出预计到 2004 年底，Google 集群中的 PC 机台数会超过 18,000 台，外存储器容量达到 5PB。在 2000 年，Google 集群中的 CPU 个数（每台 PC 机中仅有一个 CPU）只有 4,000 个，2003 年初它便增加到 30,000（每台 PC 机中有两个 CPU），因此有理由判断如今 Google 集群中的 CPU 个数可能达到或者超过 40,000 个。为了获得 Google 集群的最新信息，在本文写作过程中，我曾通过电子邮件向 Google Fellow Urs Hölzle 博士（文献[4]的通信作者）询问 Google 集群当前的 PC 机数目、PC 机具体配置等问题，但没有得到回复。以往向国外学者询问论文所涉及的问题，一般都能很快得到详细答复。本文写作过程中，我曾在 IEEE, ACM, ScienceDirect 等专业科研论文数据库中进行多次检索，以及在 Yahoo, Google 等通用 Web 搜索引擎中进行搜索，还查看了 Google Lab 主页中列出的论文清单，但获得的最新报道仅仅是 2004 年 4 月发表的文献[5]。因此，不妨可猜测：Google 集群中的 PC 机数量、PC 机具体配置信息可能是搜索引擎公司的重要商业秘密，不便公开。

文献[3]早在 1998 年就介绍了 Google 搜索引擎的逻辑结构如图 1 所示。文献[3]是至今

关于 Google 搜索引擎逻辑结构描述最为清晰的一篇论文，并且该文献中作者明确表示这也是他们阅读范围内第一篇详细描述 Web 搜索引擎结构的论文。由于图 1 仅是 Google 搜索引擎的逻辑结构图，我们根据图 1 判断服务器采到底是采用 SMP 构架的服务器，或者是 MPP 构架的服务器，还是集群服务器。图 1 对理解搜索引擎的逻辑功能模块和检索操作流程很有帮助，对我们在后面分析系统的并行性有帮助。

图 1 中同一个功能模块如果出现多次，如 crawlers，则表示该模块存在多个并行的实例。多个并行实例既可以理解为一台机器中的多个进程或线程，也可以理解成在多台 PC 机上执行的进程。以 crawlers 为例，在 Google 原型系统设计之初，可能就是同一台机器上多个线程，而在 2000 年以后可能就是多台 PC 机上都在运行 crawlers 程序。文献[3]中对每个功能模块都进行详细说明，感兴趣者可直接阅读文献[3]。

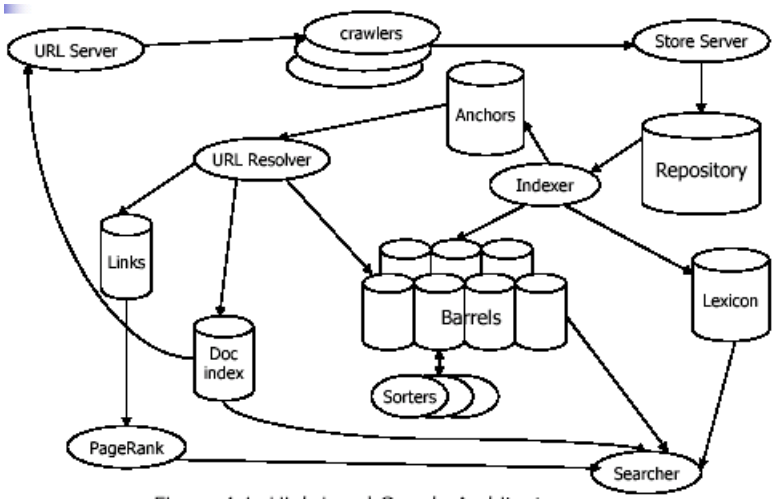

图 1、Google 的逻辑结构图

图 2 是最新的关于 Google 服务器系统结构的介绍[4],该文通信作者 Urs Hölzle 博士由于对 Google 公司的发展作出了突出贡献，被授予 Google Fellow 荣誉称号。文献[4]是我们收集到的资料中唯一专门讨论 Goolge 集群的系统结构的论文。从 1998 年到 2003 年，时间相隔五年，但比较图 1 和图 2 却发现二者并没有本质区别。图 1 中的系统结构五年后依然在更复杂、数据和计算更密集的应用环境中继续使用，这验证了图 1 中的 Web 搜索引擎逻辑功能结构具有很好的可扩展性。

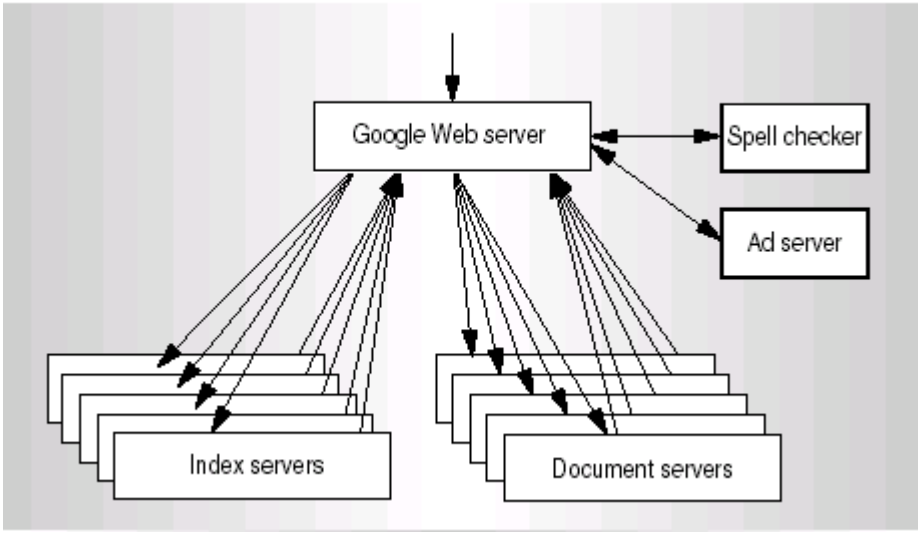

图 2 Google 的逻辑结构图

关于Google集群的物理结构最早的全面介绍可能是2001年出版的计算机系统结构经典

教材《Computer Architecture: A Quantitative Approach》一书[6]。文献[6]中专门把 Google 集群作为大型集群的综合性实例，从 PC 机内部配置，PC 机在机柜中的布局、集群系统的能耗和散热处理、网络交换设备、存储设备等多个方面进行介绍。由于文献[6]在计算机系统结构教科书中的权威地位，它的出版使 Google 集群得到了大家的广泛关注。由于文献[4]发表时间比文献[5]晚两年，以下用文献[4]中公布的 PC 机配置信息来估算 Google 集群的计算性能，以及它在 TOP500 中的排名位置。

浮点数峰值速率是大型机设计、开发和评价的一个重要指标。TOP500 在对大型机进行性能排名时有两个重要指标，一个是系统的理论浮点数运算速率，另一个是实际的 Linpack 测试浮点数运算速率[7]。文献[6]中指出每台 PC 机中只有一个奔腾 800MHZ 的 CPU，文献[3]中报道在 2002 年底一台 PC 机中有两个 2G Intel Xeon 的处理器。15,000 台 PC 机则意味着 Google 集群中有 30,000 个并行的 CPU。从因特网上不难查到 2GHZ Xero 处理器每个时钟周期内可完成两次浮点数运算，因此可得到 2002 年底 Google 集群的理论浮点数运算速率为 2G×2×30,000=120,000Gflops=120TFlops。对比 2002 年 11 月份的全球 TOP500 排名列表[7]，发现其中排名第一的地球模拟器的峰值速率是 40,960Gflops，Google 集群的峰值速率是地球模拟器的近三倍。由于地球模拟器采用向量计算机技术，我们不放把 Google 集群与 TOP500 中排名第二的 ASCI Q 作比较。

ASCI Q 是 NASA(National Aeronautics and Space Administration)第五台高性能计算机，在参加 2002 年 11 月份的 TOP500 测试时它的基本配置是 4096 个 1.25GHZ 处理器，Linpack 测试速度 7727GFlops，峰值速率为 10240Gflops [9]。ASCI Q 在 2002 年没有还没有完全开发完毕，它计划在未来达到 11,968 个处理器、12TB 内存以及 600TB 的磁盘容量的规模。比较 2002 年底的 Google 集群与 ASCI Q,发现 Google 集群的处理器个数几乎是 ASCI Q 的八倍，且每个 CPU 的主频较高，理论峰值速率 Google 集群是 ASCI Q 的十二倍。

鉴于 Google 集群是 Google 公司自己搭建，且没有采用如 MyriNet 价格昂贵通信效率高的大型机专用高速通信网络设备互连。由于无法收集到 Google 集群作 Linpack 测试的数据，但可以基本肯定的是它作 Linpack 测试的可实际运用速率与峰值速率之比要低于地球模拟器和 ASCI Q。因此在 Google 集群上作 Linpack 测试的实际速率不一定比 ASCI Q 高，但由于它的理论峰值太高，即使可利用率低至 0.5%（观察 TOP500 中的记录，可知该数值一般都大于 30%），它依然在 TOP500 的前一百名之内（因为第 100 名机器是 558GFlops）。

本文写作过程中，我查阅了 2000 年以来 TOP500 的十次排名记录，都没有发现 Google 集群。参与 TOP500 排名的机器是遵循自愿原则（可以在网上填表），可以推测 Google 有进入前 100 名的实力，却没有上榜，其最大可能就是没有参与 TOP500 排名。Google 集群不愿意公布性能数据，从很难检索到关于它系统结构分析的论文也是一个反映。

因此尽管在 TOP500 中没有发现 Google 集群，但无需质疑的是它无论是在浮点数计算能力还是存储容量方面都是世界上最顶级的计算机之一。某些大型机有很高的性能，且有很高的技术含量，却没有出现在 TOP500 名单中的情况其实在我国也有。如众所周知的神威巨型计算机，它不仅具备很高的峰值速率，也具有良好的实际性能，至少在国内它与曙光公司、联想公司开发的巨型机具有相当的性能，然而在 TOP500 上一直没有出现它的名字。

国外关于利用集群技术开发大型机的 Beowulf 论坛中已有人提出：设计大型机，并注重它能否进入 TOP500 以及在 TOP500 中的名次只是学术界的一个游戏而已[9]。利用集群构架开发的大型机，进入 TOP500 其实不能代表太高的技术水平，这应该是近年来关于大型机开发的一个共识。如今国内的大型机开发技术已经达到较高的水平，未来几年无需再把能否进入 TOP500 前十名当作十分重要的目标。因为我们过去设计的一些大型机由于应用效率不高，已经造成了不少资源闲置现象，关于该类报道在互联网上比较多。

Google 集群完全能进 TOP500，但它却不参加排名。Google 集群能否进入 TOP500，丝毫

不影响大家对它高性能、高可用性和高可扩展性的怀疑。作为集群系统的典型例子进入教科书，就是对 Google 集群的极大肯定。文献[3]中作者再三强调 Google 集群是由 Google 公司自行开发的集群系统，并不采用最先进的技术（一般最先进的技术都比较昂贵）。Google 集群设计和维护中无处不体现强烈的追求最大性价比、实用的特点。Google 集群的设计者不一次性购买很多资源，而是不够时再投入。总之，Google 集群构建和维护过程中始终坚持的实用主义特点，值得我们在大型机设计和购买过程中学习。

3、Google 的文件系统

作为一个巨大的数据检索系统，Google 并没有采用常见的数据库管理系统来管理 Web 文件及相关数据。Google 集群如今收集了 80 多亿个 Web 文档，平均每天执行两亿多次用户查询请求。文件数目太多，则检索速度慢；且每个文件大小相差甚远，容易导致硬盘文件存储管理费用巨大。文献[3]中介绍，Google 把多个 Web 文件汇集到一个巨大的文件中进行存储、管理，当然对一个大文档有相关记录信息。关于大文件说明信息的格式如图 3 所示[3]。另外因为 Web 文件数量太多，所占存储区间太大，Google 采用 ZLIB 压缩算法先对原始 Web 页面信息进行压缩，然后只存储压缩后的结果。ZLIB 压缩算法对 WEB 文档的压缩比为 3：1，因此一个 64MB 的大文件，实际上包含 192MB 的原始 Web 文档。

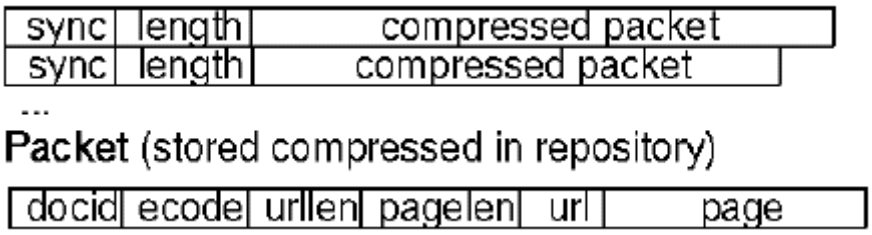

图 3 Web 文件的描述信息

Google 集群中的硬盘是普通的 IDE 接口的 PC 机硬盘，虽然便宜但可靠性较低。Google 集群维护人员在工作中发现，存储器出现故障在 Google 集群中并不是少见的异常现象，而是频繁发生。三万个硬盘(15,000 台 PC 机，每台机器中有两个硬盘)同时工作，都不出现故障，用概率计算的方法一算就知道几乎是不可能事件。Google 集群总在频繁的执行文件读操作，硬盘出现故障的可能性就更大。硬盘常出故障的现实环境与传统分布式文件系统设计的基本假设有冲突。Google 集群应该设计具备持续监视、错误检测、容错和自动恢复功能的分布式文件系统。

Google 集群运行过程中还发现了许多来自应用、操作系统、用户、存储器、网络等多方面存在的软件故障。Google 集群中的文件操作有一个特性：每个文件基本上是只执行一次写入操作，以后都是频繁执行只读操作，随机写操作几乎不发生。当 crawlers 从网络中采集到一个 Web 文件，在内存中完成相关处理后，就写入到一个大文件中，以后每次查询只是执行一次数据读取操作。银行、电子商务等常见的大型数据管理系统中则没有这个特性。针对来自应用领域的特征，Google 公司的研发人员在分布式文件系统开发中采用了不同于传统分布式文件系统 I/O 操作的假设。

Google 开发了公司内部的分布式文件系统 Google File System（GFS），该文件系统的系统结构如图 4 所示。GFS 中包括单一的 GFS Master，多个 GFS chunkserver。文献[3]和文献[10]发表时间相差五年，但不难判断文献[3]中的大文件就是文献[10]中的 chunk。图 4 表示 GFS 不是完全在操作系统层设计的分布式文件系统，而是在已有的 Linux 文件系统之后再封装的分布式文件系统，因此它依赖传统的 Linux 文件系统。GFS master 中存储了三种重要的元数据：文件和 chunk 的命名空间、文件到 chunk 的映射表以及 chunk 及其备份的位

置信息（为了提高文件存储的可靠性，GFS 中对同一份数据一般存储在三台不同的主机上。对一个 chunk 而言，它就有两个分布在其它 PC 机上的备份）。GFS chunkserver 中每个 chunk 大小为 64MB，关于每个 chunk 大小的具体数值选择并不是随意的，而是很有技巧。关于 64MB 大小的 chunk 设置的优点和缺点在文献[9]中有很详细的讨论。

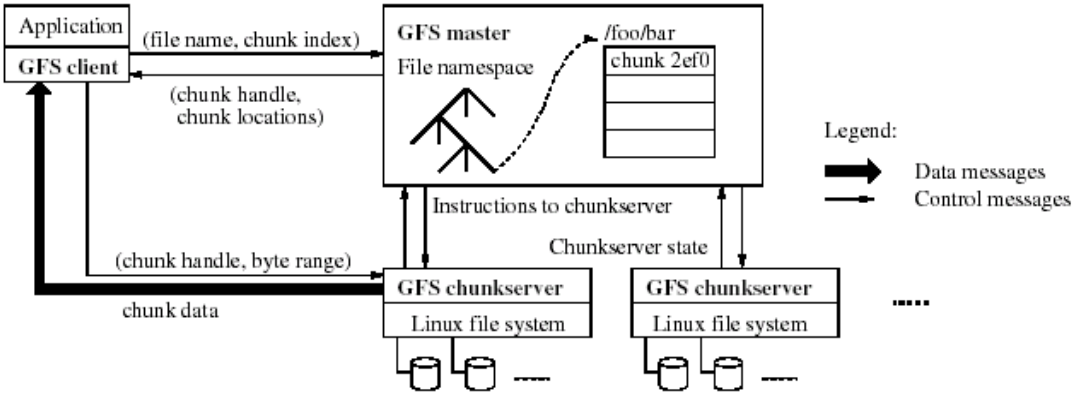

图 4 Google 文件系统的结构

GFS 中执行一个简单的读数据操作工作流程如下：首先，应用客户端把文件名和文件在一个 chunk 中的偏移量转换成一个包含该文件数据的 chunk 索引；然后，客户端向 GFS master 发送请求，请求中包括所需要的文件名以及 chunk 索引。当 GFS master 收到请求并通过 chunk 映射表映射后，它向客户端作出响应，反馈给客户端相应的 chunk 句柄以及该 chunk 备份的位置。客户端收到反馈信息后，将以文件名和 chunk 索引为关键词进行缓存。有了所需文件 chunk 索引和位置信息，客户端一般从多个 chunk 服务器中选择一个离自己最邻近的 chunkserver 发出数据访问请求。一般数据访问存在空间局部性，以后如果该应用客户端需要访问同一 chunkserver 中的数据，它不再向 GFS server 发送请求，而是直接向 chunk 服务器发送数据读取请求。关于 GFS 更详细的讨论在此不再进行更深入讨论，如果希望进一步了解可以参考文献[9]。

4、Google 的存储系统

Google 集群收集到的 Web 页面就已达到 80 多亿张，另外它还有十亿多张图片。文献[5]中报道 Google 集群中的数据容量在 2004 年就达到了 5PB。作为一个容量巨大且访问频繁的存储系统，Google 却没有采用流行的网络存储技术和附网存储技术，甚至就连比较廉价的磁盘阵列也没使用。Google 采用的存储方法是用最常见、最普通的一个 PC 机中带两个硬盘的存储方式。Google 放弃主流的并行海量数据存储技术，采用健壮性较低的 PC 机带硬盘存储的方式最主要的原因是希望降低成本。虽然这种存储方式不太可靠，但通过 GFS 可实现高效、可靠存储，已被实践检验是有效的海量存储解决方法。

2003 年市场上最常见的 PC 机硬盘存储容量是 80G。根据文献[4]中报道，每台 PC 机中配有两个 80G 的硬盘，不难计算出 Google 集群的总容量是 80G×15,000×2＝2,400,000G＝2.4PB。由于可靠性和可用性对 Google 十分重要，不是所有的存储区间都用来存储有效数据。为了提高可用性，有一半的硬盘是作镜像存储。当一个硬盘出现故障，或者数据读写错误时，可用另一个硬盘启动系统或访问其中的数据。集群中的每台 PC 机都存储了一份独立完整的操作系统，又有一部分存储区域去存放系统软件，因此实际有效存储空间比 1.2PB 还小。同样出于价格成本因素考虑，Google 集群中没有使用 SCSI 接口的硬盘。使用 SCSI 接口的硬盘，不仅硬盘自身价格高，且还需转接卡。为了提高数据读取速度，Google 集群中的硬盘转速比较高，从而降低旋转等待时间。

2003 年底，Google 中每台 PC 机中竟然配置了高达 2GB 的内存。由于每个 chunk 高达 64MB，使用大内存对提高系统的性能有良好效果。曾对存储器容量对系统性能的影响进行测试，发现存储器系统不是整个系统性能的瓶颈。

5、Google 的可靠性和可用性

可靠性是指系统正常运行时间，可用性是系统正常运行时间与正常运行时间以及故障维修时间之和的比率[11]，高可用性是评价超级计算机和服务器系统的关键指标之一。作为商业网站，Google 集群必须能保证每周 7 天、每天 24 小时都正常工作。即使是在系统维护操作时，也不能让整个系统停机。一万多台 PC 机，总会存在各种各样的故障和异常，几乎所有故障都会降低系统的可用性。即使系统不出故障，Google 集群中的 PC 机每两年也会更新一次。PC 机更新是在硬件上彻底更换，更新的过程必定会终止一部分机器服务。设备更新比较耗时，15,000 台机器同时更新显然不是两个小时内可以完成。以下分析 Google 集群提高系统可用性的方法。

首先，Google 是 Web 搜索引擎，可用性要求比较弱。搜索引擎的可用性需求与银行、电信业务中的可用性存在区别，这些区别导致了不同的可用性维护策略。对银行业务连续执行同样的请求，得到的响应必须一致。但对 Web 搜索引擎，用户期望的可用性是只要能访问 Google 的入口，键入检索词，几秒钟后能得到检索结果即可。2004 年 4 月 17 日,我输入“华中科技大学”一词进行检索，得到了 1,170,000 个检索结果；而在 4 月 21 日执行同样的检索，我只得到了 917,000 条结果。显然两次检索结果差异巨大，但实际上普通用户很难发现这种差异，甚至根本就没考虑过这种现象。

Google 中对 Index 和 Web 页面数据都采用分布式存储。每台 PC 机都可能出故障，出现故障后需要一定的故障维修周期。在维修周期内，该机器中所存储的数据可能就不能正常访问，因此导致反馈给用户的检索结果减少。因此，连续多次执行同样的检索请求，但检索结果不一致现象就能得到合理解释。这种非严格的可用性，给 Google 的维护以及技术方案选择提供了宽松的环境。相反，银行则不得不采用昂贵的硬件设备来维持系统的严格可用性。

其次，Google 通过分布在不同地区的多个镜像站点来提供系统的可用性。Google 集群耗电量巨大，占地面积大，停电是导致系统可用性降低的致命因素。Google 公司位于加州，加州近年来出现了能源短缺、电力供应紧张现象，近年来加州曾多次出现大面积停电现象[12]。单机或服务器保证电力供应方法是采用 UPS 电源，而 Google 集群由于功耗太大，估计采用 UPS 的可能性比较小，公司自备发电机发电可能是更可行的方法。Google 仅是一个 Web 搜索引擎，即使是公司内部有备用电源，也不能保证当地区停电时网站正常提供网络服务功能。原因是 Google 与外界连接的通信网络也因停电而终止工作，因此无论 Google 集群的可用性有多高，只要通信网络具有单一性，它依然无法保证高可用性。

另外，地震、恐怖袭击等其它灾难性事故也将导致单一 Google 集群无法保证高可用性。加州是太平洋板块和美洲板块的交界处，地震活动比较频繁。2005 年 4 月 16 日 12 时 18 分在加州 WSW of Mettler 发生了 5.1 级的地震[13]；2003 年 12 月 23 日在加州发生了里氏 6.5 级的地址，导致多人死亡。运行在地震频繁地区的大型机有必要考虑地震对整个系统可用性的影响，实际上 Google 公司的研究人员也考虑了该问题，并采取了相关对策。

2001 年 Google 有三个镜像站点，两个分布在加州的硅谷，另一个在美国东海岸的弗吉尼亚。每个 Google 站点都采用 OC48 (2488Mbit/sec)的带宽连接到因特网，硅谷中的两个临近站点还用一根 OC12 的光纤互连，以便紧急情况下或网络故障时两个镜像站点可共享一根 OC48 光纤连接互联网。弗吉尼亚州的那个站点在 2003 年也有了自己的镜像，其连接方法就和硅谷中两个镜像站点的互连方法一致。从一个站点变成地理位置上分离的四个站点，软硬件设备的成本就增加了三倍，但可用性几乎达到了 100%。中国用户访问 Google 经常不通，

估计是通信网络故障原因较多。这些来自网络中的技术的与非技术处理是 Google 公司无法解决的。

高度冗余提高了 Google 的可用性，但 Google 公司对镜像站点的投资并不是真正的冗余投资。服务器常见的冗余方式是双机备份，当一台机器出现故障时另一台机器立即接管故障机器上的任务。简而言之，就是两台服务器实际上只具备一台服务器的工作效率。如果服务器工作稳定，有一半投资是冗余。由于搜索引擎对可用性要求不是很严格，且每个 Google 站点自身有很高的可用性，Google 的镜像站点不是另一个站点的热备份。通过基于 DNS 的负载均衡方法，DNS 服务器把域名地址 www.google.com 解析成不同的 IP 地址，把查询请求分配到四个镜像站点。基于 DNS 的负载均衡方法并不是 Google 首创，它是计算机网络技术中的一个常见技巧，简单高效。如 DNS 服务器收到一个来自中国大陆地区的搜索请求，只要加州的两个服务器不是特别繁忙，它就把域名地址解析成加州两个站点中较为空闲的一台。请求在加州处理，降低了网络通信中的线路传输延时，对降低响应时间有好处。同理，来自欧洲用户的请求则解析到弗吉尼亚的镜像站点。四个站点中的某个站点出现故障或过于繁忙，DNS 服务器可以暂时不把请求分配到该站点，而是转移到其它的三个站点，因此 Google 看来是利用冗余的方法来提高系统的可用性，但建立镜像站点实际上几乎算不上冗余投资。这从一贯坚持节省投资成本的 Google 而言，无疑是一个很好的设计方案。

第三、Google 坚持采用软件方案来提高可用性，而不是使用硬件技术。服务器和大型机中的冗余一般是用硬件，硬件方式意味着更多投资。Google 公司的维护人员可以开发软件，如 GFS，采用软件方法可达到提高可用性的效果。Google 是最大的 Linux 用户，它可把应用需求直接向 Red Hat 公司提出，让对方的设计人员开发面向 Google 的操作系统[14]。

6、Google 如何实现高并行性

如何提供系统的并行性是大型机设计和开发的关键问题，并行性高则意味着系统的加速比大，有利于系统的可扩展性。系统的加速比与实际应用自身的特性紧密相关，如果应用中计算部分所占比例小，而多处理器之间的通信、同步操作多，则并行性难以提高。Google 集群的设计者在多篇论文中都表明 Google 集群的处理能力与 PC 机数量呈线形关系[4]，显然这是并行计算中所期望的理想状态。本文认为 Google 集群提高并行性的技术可归纳为宏观意义上的多发射、多流水。

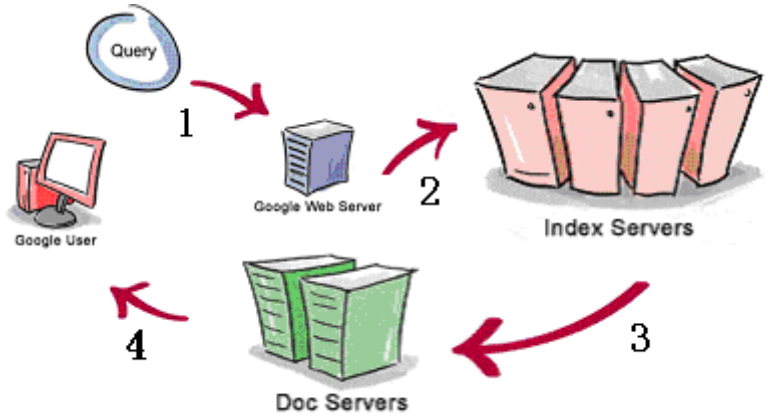

图 5 请求处理流程

图 1 示出 Google 的三大功能模块：负责 Web 页面收集的 crawlers、负责 Index 检索的 Index server、以及负责文档检索的 Doc Server。Crawlers 只负责从网络中不断的采集网页，它与用户的检索请求没有直接联系；Index 以及页面检索是直接为用户的请求服务。三者之间通信和同步操作少，适合并行工作。Crawlers 以固定周期去发现和采集网络中的 Web

页面，1998 年、2000 年、2001 年以及 2003 年的文献关于采集周期的报道都是每一个月更新一次。每个 crawler 所采集页面的地址是由系统分配的，因为分配的地址不同，多台执行 crawler 程序的 PC 机可以并行操作。只要 Google 集群连接到网络的带宽不存在瓶颈，则数据采集信息处理能力与 PC 的数量成正比增长。

一个请求在 Google 中的处理过程如图 2 和图 5 所示[15]。步骤 1 是用户通过 Web 页面将检索词发送到 Google Web Server。步骤 2 是 WWW 服务器将查询发送到索引服务器。目前在步骤 2 执行之前，Google 做了一些小处理，如：判断检索词的英文拼写是否正确，如果出错向用户给出提示信息。Google 盈利的主要来源之一是广告业务，因此在步骤 2 执行之前，还有一些关于附加广告操作。

Google Web Server 对每个请求所作处理较少，一般不是系统的瓶颈。Index 检索与 Doc 检索之间存在严格的时序关系，即对同一个请求只有在 Index server 处理完成后，才能去访问 Doc Server。把一个请求分成两次查询主要是从效率角度考虑，因为原始 Web 页面有 80 多亿张,直接检索则范围太多，检索时间长。Google 先对容量大小为数 TB 甚至是几个 PB 的原始数据建立高达数 MB 的索引描述信息，根据 Index 检索结果，只对部分 Web 页面进行搜索。Index 检索完成后，Index 服务器不再处理该请求，而是直接去处理下一个请求。当 Index 服务器在为第 n 个请求作检索的时候,Doc 服务器可能是在为第 n-1 个请求进行搜索。Index 服务器和 Doc 服务器在同一时刻并不处理同一个请求，二者可并行为用户服务。若把每一个宏观检索类比为流水线中的功能部件,那么 Index 服务器和 Doc 服务器是流水线执行。

多个 Doc 服务器可以得到数以百万计的检索结果，如本文中描述的对“华中科技大学”检索。多个结果在反馈给用户之前，必须进行一次信息汇总和结果评价排序操作，最后将最好的结果在前几十条记录中反馈给用户。

多发射技术在 Google 中也有应用。从站点层次来看，Google 有多个镜像站点，每个镜像站点之间可以并行为用户服务。位于美国东海岸的镜像站点为美国东部和欧洲用户服务，而加州的镜像站点并行的为亚洲及美国西部用户服务。从站点内部来看，Google Web Server 并非是一台 PC 机，它肯定有多台 PC 机负责该操作。总之，在同一时刻 Google Web Server 可以接受多个用户请求，并可以向 Index 发出多条查询任务。类比单个芯片内部存在多条流水线的现代 CPU 系统，我们可以认为 Google 集群中有多条流水线，是一个多发射系统。

表 3 每秒 Google 处理的请求数和 Google 中的页面数量

时间	Web 页面数	每天处理的请求数(百万)
1998 年		0.01
1999 年 1 月-6 月		0.5
1999 年 8 月-11 月		3
2000 年 5 月-6 月		18
2000 年 11-12 月	1.3Billion	60
2001 年 1 月-2 月		100
2001 年 11 月-12 月	3Billion	
2002 年 7 月-8 月	2.4Billion	
2002 年 11 月-12 月	4Billion	
2004 年 1 月-2 月	4.28Billion	
2004 年 11 月-12 月	8Billion	
2005 年 4 月	8Billion	200[2]

互联网的普及导致 Google 在单位时间内处理的请求越来越多。表 1 示出了自 1998 年以来，Google 平均每天处理的检索请求数目增长情况。表 1 中的数据来自 Google 公司提供的介绍信息[15]。关于 2005 年 4 月份平均每天处理的用户请求数是超过 2 亿，即大于 200M。

7. 结束语

Google 是世界上最成功的网络搜索引擎，它在创办之初就发现并坚持了一个十分正确的搜索结果评价标准:精确率高于查全率，并设计了比较科学的网页评价算法。在 Google 最具原创性的两篇论文中，对来自硬件领域的可扩展性、可靠性、可用性根本都未曾涉及。Google 公司创始人在攻读博士期间发表的论文主要是讲述 Google 的软件体系、数据结构和页面评价算法，由此可判断他俩擅长软件设计，而不是偏向硬件的计算机系统结构。

随着 Belwolf 的成功，集群技术成为今年来国际上大型机和高性能计算领域的主流技术。Google 集群的设计者在多年前就巧妙的利用集群技术，无疑是一个十分正确的技术选择。因为没有哪台大型机能以如此低的价格，提供如此强的可扩展性。

Google 是世界上应用效率最高的集群系统之一，但对集群技术自身的发展它并没有作出太多贡献。Google 集群在某种程度上来看，也许不能成为严格的集群服务器，甚至更像一个巨大的计算或存储局域网。因为集群系统研究中最关键的技术是单一系统映象,而所有关于 Google 集群的介绍中都没有涉及。估计 Google 集群是通过网络的方式来进行负载均衡和数据访问，不具有严格的集群定义。随着人们对集群定义的逐步放松，特别是 Patterson 教授和 Hennessy 教授的《计算机系统结构：一种量化的研究方法》中用很大篇幅详细介绍 Google 集群之后，它就变成了大型集群成功应用的典范。Hennessy 教授既是 Stanford 大学的校长，又是 Google 公司的董事。作为计算机系统结构领域的国际著名学者，他有可能直接参与并指导了 Google 集群的设计和开发,在目前所有关于 Google 集群的文献中,文献[6]是最全面和细致的一篇。Google 集群能否称得上严格意义上的集群，在国外也有争议。但无论是否是集群，对 Google 而言没有太多意义。因为它已成功的解决了一个大计算量、大存储量、高实时性的应用需求，并且多年来工作得很好。

总之,从计算机系统结构角度来看，我最推崇 Google 集群研究者和维护者的观点：用最少的钱、用最成熟的技术去构建稳定、高性能、高可用性的系统。目前国内也有很多耗费巨资购买的服务器或者大型机，而真正得到高效应用较少。有些计算或存储服务需求比 Google 要宽松得多，为什么我们不改变思路，用成熟的技术去构建一个自己的大型计算机系统？本文对 Google 集群的评价是：用成熟、廉价的技术去构建了一个先进、卓越的计算机系统。这种实用主义的态度，值得我们未来在计算机硬件系统设计和软件系统开发中学习。

致谢

本文是华中科技大学计算机学院谢长生教授讲授的《高等计算机系统结构专题》课程的课程论文，感谢他提供了 Google 集群系统结构分析这个有趣的问题、并深入浅出的讲授了计算机系统结构领域的技术进展及思维方法。

参考文献

[1] The Google Timeline, http://www.google.com/corporate/timeline.html.

[2] The World's 500 Most Influential Brands, http://brand.icxo.com/brand500/top500_1.htm.

[3] Sergey Brin and Lawence Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," Computer Networks, Vol. No. 1998, pp. 107-117.

[4] Luiz André Barroso, Jeffrey Dean, and Urs Hölzle, "Web Search for a Planet: The Google Cluster Architecture," IEEE Micro, Vol. 23, No.2, March 2003, pp.22-28.

[5] Chris Mellor, Google's Storage Strategy, http://www.techworld.com.

[6] John L. Hennessy, David A. Patterson, "Computer System Architecture: A Quantitative

Approach, (3rd edition)", Morgan Kaufmann, May 15, 2002.
[7] TOP500 supercomputer, http://www.top500.org.
[8] List of November 2002, http://www.top500.org/list/2002/11/
[9] Google's cluster, http://www.beowulf.org.
[10] Sanjay Ghemawat, Howard Gobioff and Shun-Tak Leung, "The Google File System," Proceedings of the nineteenth ACM symposium on Operating systems principles, October 2003.
[11] Kai Hwang and Zhiwei Xu, "Scalable parallel Computing, Technology, architecture, programming," WCB/McGraw-Hill, 1998.
[12] http://tech.sina.com.cn/i/w/65936.shtml.
[13] Recent Earthquakes in California and Nevada, http://quake.usgs.gov/recenteqs/index.html.
[14] Red Hat Linux Powers Google's Award-Winning Search Engine, Business Wire, May 30, 2000.
[15] Press Center of Google, http://www.google.com/press/descriptions.html.