**北京航空航天大学计算机学院**

**硕士学位论文文献综述**

**论文题目**：面向实时检索的搜索引擎缓存技术研究

**专 业**：计算机技术

**研究方向**：信息检索

**研 究 生**：欧韬

**学 号**：ZY1306107

**指导教师**：李建欣 王丽宏 马宏远

**北京航空航天大学计算机学院**

2014年11月25日

目 录

[**摘要** 2](#_Toc405217322)

[**Abstract** 2](#_Toc405217323)

[1. 概述 2](#_Toc405217324)

[1.1. 背景概述 2](#_Toc405217325)

[1.2. 阅读文献的概述 4](#_Toc405217326)

[2. 搜索引擎缓存技术研究现状与发展趋势 4](#_Toc405217327)

[2.1. 搜索引擎用户查询日志分析 5](#_Toc405217328)

[2.2. 面向有限容量缓存和固定索引的缓存技术研究 6](#_Toc405217329)

[2.3. 面向无限容量缓存和持续更新索引的缓存技术研究 7](#_Toc405217330)

[2.3.1. 缓存更新技术 7](#_Toc405217331)

[2.3.2. 缓存内容失效技术 8](#_Toc405217332)

[2.3.3. 生存时间机制 9](#_Toc405217333)

[2.4. 信息检索技术 9](#_Toc405217334)

[3. 结论 12](#_Toc405217335)

[4. 主要参考文献 13](#_Toc405217336)

**面向实时检索的搜索引擎缓存技术研究**

# **摘要**

本文首先介绍了搜索引擎缓存技术研究现状、发展趋势和面临的问题。从SIGIR,WWW,CIKM等信息检索领域顶级国际会议的论文来看，在搜索引擎缓存技术主要有两个研究方向。第一：面向有限容量缓存和固定索引的搜索引擎缓存技术研究；第二：面向无限容量缓存和持续更新索引的搜索引擎缓存技术研究。因而本文接着讨论了不同的搜索引擎缓存管理机制，比较了它们的特点。本文介绍了相关论文中提出的一些主要的搜索引擎缓存技术，并且指出这些技术的一些不足。

**关键词**：实时检索、搜索引擎、缓存管理

# **Abstract**

This paper first introduces the related work, development trend and problems of search engine cache technology. Papers of top international conference such as the SIGIR, WWW, CIKM, who lead information retrieval fields show that there are two main research directions in the search engine cache technology. Firstly: Search engine caching technology research for the limited capacity of the cache and fixation index; secondly: Search engine cache technology for the infinite capacity cache and considering continue updating of the search engine index. This paper then discusses the different search engines cache management mechanisms, compares their characteristics. We introduced the major search engine caching technologies in related paper and pointed out the some drawbacks of these technologies.

**Keywords**: real time retrieval, search engine, cache management

# 概述

## 背景概述

据中国互联网信息中心（CNNIC）研究报告显示，截止到2014年7月，中国网民达到6.32亿，其中Web搜索引擎用户规模为5.07亿，（以下简称搜索引擎）使用率为80.3%，搜索引擎已经是人们获取信息的主要工具。一方面，互联网网民增多，搜索引擎工作负载变大；另一方面，互联网网页也逐年增多，其规模已经超过1500亿，搜索引擎需要尽可能多地索引这些网页以提供准确和丰富的查询结果。在如此海量信息中，高效获取所需查询对学术界和工业界提出挑战。

搜索引擎通常包括两个主要部分：第一部分执行离线操作，执行网页抓取（Crawler）和根据网页文本内容构建索引（Indexer）；第二部分执行在线操作，处理来自用户的查询请求（Query Processor）。从用户的角度来看，其希望达到如下目标：（1）快速查询响应时间；（2）丰富查询结果；（3）高质量查询结果。由于搜索引擎具有如此大量的用户以及如此之大的网页规模，从服务器的角度来看，其需要提供如下能力：（1）高查询效率；（2）高查询吞吐量；（3）海量信息索引；（4）高质量查询排序。典型的商业搜索引擎采用分布式架构，每个集群内部能够独立完成一个查询，通过复制多个集群以达到能够支持特定用户规模的查询吞吐量。降低集群规模，即减少服务器数量可以减少运营成本和能源消耗。在保持同等服务质量的前提下，意味着需要提高单个集群的吞吐量，单个集群的吞吐量是搜索引擎检索系统效率的体现。检索系统效率的优化对于减少运营成本、降低能源消耗有着重要意义。

巨大的商业价值和重要的学术价值吸引了来自于业界和学术界不同领域的研究人员在搜索引擎检索系统效率优化这一问题上开展研究。自本世纪初以来，互联网步入高速发展时期，一些搜索引擎检索系统效率优化方法[4-6]先后被提出，主要有并行处理（Massively Parallel Processing）、索引压缩（Index Compression）、索引裁剪（Index Prune）、倒排表跳转（List Skipping）、提早结束（Early Termination）等技术。除此之外，缓存技术（Caching）作为重要的系统效率优化手段而被搜索引擎检索系统采用，信息检索领域的针对搜索引擎环境下的缓存研究工作也成为搜索引擎系统结构方向的研究热点，信息检索以及互联网领域顶级会议SIGIR、WWW每年都会有相关的理论成果发表，部分研究成果还被提名角逐最佳会议论文。可见，缓存技术在信息检索学术界研究与工业界应用都具有举足轻重的地位。

缓存技术是一种通过存放将来可能被请求的数据以提高数据访问速度。被存放的数据可能是计算结果或者数据的副本。如果请求的数据在缓存中被查找到，就直接获取该结果或者副本，否则，请求需要再次计算或者从其位置获得该数据。如果更多的请求能够被缓存所满足，则意味着系统的整体效率会被提高。缓存被广泛应用于计算机系统及应用软件的各个层次，譬如CPU缓存、磁盘缓存、Web缓存。

Web2.0时代的到来给网络搜索引擎缓存技术带来了新的挑战，网民对信息的实时性有了更高的需求，如方便用户查找筛选新闻、微博等时效性高、更新速度快的文档，这就对搜索引擎的实时性提出了更高的要求。搜索引擎为了更好地满足实时检索需求，需要频繁地更新倒排索引，会引发搜索引擎缓存页面的频繁失效，影响搜索结果的实时性。搜索引擎缓存难以判断倒排索引的某次更新是否影响了查询结果而使缓存里的查询实际与当前索引计算出的结果不一致。

对搜索引擎缓存技术的研究成果能够被商业搜索以及开源搜索引擎所采用，提高其检索系统效率。尤其对于面向对大规模Web环境的搜索引擎检索系统具有重要应用价值。搜索引擎结果缓存的优化能够提高搜索集群的效率，进而可以减少搜索集群的数目，从而节省硬件资源，降低能源消耗。

本文的工作正是在上述环境下展开的，目的是要寻求一种高效的搜索引擎缓存机制。

本文余下的部分我们将在1.2节概括相关文献中搜索引擎缓存技术的研究方向；在第2节总结搜索引擎缓存技术研究现状与发展趋势，同时探讨了不同背景下的搜索引擎缓存技术；在第3节总结回顾了搜索引擎缓存技术；在4节提供了主要的参考文献。

## 阅读文献的概述

阅读的国外文献，以SIGIR,WWW,CIKM等信息检索领域顶级国际会的论文为主。许多文献的作者如Cambazoglu, Junqueira等人来自雅虎研究院。国内的文献来自中文信息学、计算机研究与发展等。其研究方向主要分为两个阶段：一阶段，2010年以前对搜索引擎缓存的研究大部分假设物理缓存具有有限的容量，而搜索引擎的倒排索引保持不变；二阶段，2011年开始，随着twitter, facebook等实时信息传递系统的流行，以雅虎研究院为代表的研究人员开始关注实时检索下的缓存技术。

# 搜索引擎缓存技术研究现状与发展趋势

搜索引擎缓存与CPU缓存有着不同的需求。首先,Web页面大小不固定,只有整个Web页面在缓存中才可能命中缓存。其次,从Web服务器抓取网页有些延迟。第三,大量用户生成的请求而不是一个少量的程序。

2010年以前对搜索引擎缓存的研究大部分假设物理缓存具有有限的容量，而搜索引擎的倒排索引保持不变； 2011年开始，随着twitter, facebook等实时信息传递系统的流行，以雅虎研究院为代表的研究人员开始关注实时检索下的缓存技术。

## 搜索引擎用户查询日志分析

搜索引擎用户查询日志记录了搜索引擎收到的不用用户的查询请求信息，是研究和分析搜索引擎工作负载的重要载体，各大商业搜索引擎公司查询日志作为重要战略资源。用户查询日志分析在查询推荐、用户相关反馈等研究领域被广泛采用。Cockburn[23]和Tauscher[24]等人对Web用户的浏览行为进行分析。Silversein[25]等人对商业搜索引擎的用户日志进行了大规模分析。Baeza-Yates[26]等人对英文搜索引擎Excite,Vivisimo,Fast的用户查询日志进行了分析。Xie[7]等人对西班牙文搜索引擎TodoCL的用户查询日志进行了分析。李亚楠[9]等人对中文商业搜索引擎用户查询日志进行了大规模分析和研究。马宏远等人[21]基于真实的中文搜索索引擎大规模用户查询日志，针对搜索引擎缓存应该进行了研究。

对查询日志里用户查询独一查询频率和查询利用距离的统计表明：查询频率符合Zipf定律。少数高频查询占据了大多数查询请求，在某时间段内搜狗的查询日志上的统计数据显示频率最高的20%的独一查询占据了80.45%的查询请求量，而33.67%的独一查询只被提交了一次。平均查询重复次数为3.82。用户查询还具有良好的局部性，多数查询的复用距离在1至100之间。

对查询日志查询发起的时间进行统计发现：在一天之中，夜晚用户发起的查询量远少于白天的查询量。一天内的查询量与时间关系大至符合正弦函数，在凌晨4点左右查询量达到最低值，在15点左右查询量到达最高值。在一周之中，周一的用户查询量最高，周二、三的查询量下降约20%。

用户查询的特性能够被利用来优化搜索引擎的效率和资源分布。查询的重复规律能够作为设计缓存替换、许可策略的依据。查询量较少的搜索引擎后台空闲时间能够用来预取缓存内容或更新陈旧查询结果，提升缓存命中率。

## 面向有限容量缓存和固定索引的缓存技术研究

缓存作为有效减少搜索系统响应时间和系统负载的关键技术，被广泛应用于系统性能优化。Markatos[12]比较了LRU缓存替换策略及其若干变种和静态缓存策略的命中率，得出静态缓存策略仅适用于缓存容量较小的情况。在大容量缓存上的性能很差，操作系统里使用的基于替换策略如lRU或LFU的动态缓存能够达到相对更好的性能。可以认为静态缓存利用了用户查询分布的二八定律，将经常被用户发起的那20%查询结果存入静态缓存。而LRU策略利用了用户查询的时间局部性。Fagni[13]等人设计了一个双层缓存(Static-Dynamic Caching (SDC))，包括一个存储频繁查询的静态缓存和一个采用替换策略的动态缓存。SDC同时考虑到了查询的时间局部性和查询历史的统计特性。在Altavista日志上的实验表明，SDC缓存达到了超过30%的缓存命中率。

Xie等人[29]讨论了缓存位置的影响，分别讨论了客户端缓存、代理服务器缓存和搜索引擎服务器缓存技术。Lempel和Moran[11]计算衡量查询在将来被发起概率的得分，这个技术被称为概率驱动缓存(Probability Driven Caching (PDC)). Baeza-Yates[6]等人为搜索引擎结果缓存设计了许可策略(Admission Policies)，来阻止不频繁的查询通过替换策略获得频繁的查询在缓存中的空间，并在对用户查询日志的实验中得到了较好的缓存命中率。

查询结果并非唯一能可以被搜索引擎缓存的数据。Saraiva等人[32]提出一种两级缓存方案，结合了缓存查询结果和缓存被频繁访问的倒排记录表。Long 和Suel[31]扩展了这一思想，同时缓存一对总是在查询中一起出现的词项倒排记录表的交集。Baeza-Yates[14]等人看出，缓存存储倒排记录表可以比只存储查询结果和词项获得更高的缓存命中率。应当注意的是，在包含搜索引擎、倒排记录表缓存和查询结果缓存的大规模分布式系统中，它们可能不需要在同一台机器上竞争内存资源。SKobeltsyn等人[33]的工作描述了一个ResIn体系，缓存查询结果和修剪过的索引。Zhang等人[28]通过倒排表压缩结合缓存技术，取得了较好的缓存命中率。

以上的搜索引擎缓存技术研究对缓存数据压缩、缓存位置、缓存记录活跃度等方面进行了研究，但没有考虑到搜索引擎更新对缓存内容有效性的影响。

## 面向无限容量缓存和持续更新索引的缓存技术研究

随着存储技术的进步，存储硬件设备的容量不断增大、访问速度不断提高、价格越来越便宜。与早期的工作聚焦于有限容量的缓存不同，越来越多的工作[1,2,3,4,5,8]假设结果缓存有无限的容量。Jonassen[1]等人认为廉价的存储设备容量大到足够存储所有过去用户查询的结果。在缓存容量无穷大的假设下，除了特定查询的第一次发起(compulsory missed)，其它的查询都能由缓存提供答案。然而，一个无穷大的缓存面临着缓存结果陈旧(stale)的问题。在倒排索引更新频繁，实时性要求高的搜索系统中，缓存结果的新鲜度极大地影响用户体验。

搜索引擎通常包括两个主要部分：第一部分执行离线操作，执行网页抓取（Crawler）和根据网页文本内容构建索引（Indexer）；第二部分执行在线操作，处理来自用户的查询请求（Query Processor）。爬虫从网络上获取新的或修改后的网页文档，删除不再存在的文档，从而不断地更新搜索引擎的文档集合。索引器周期性地处理文档集合并生成新的倒排文件和辅助数据结构。最后查询处理器用索引器生成的倒排索引解析用户查询。

一个网络引擎的运行时(runtime)部件通常还包括一个位于搜索引擎前端和查询处理器之是的查询结果缓存。缓存提供两个令人满意的好处：一是，它减少查询用户所察觉的时延。二是，它减少后面查询处理器的负荷。这样的缓存模块可以和查询处理器在同一台机器上运行也可以在另外的机器的运行。

然而，随着索引的变化，某些查询被缓存的结果不再反映最新的内容而变得陈旧。陈旧的查询精确的描述为：因为索引更新而使文档得分前k的查询结果改变了的查询。为了保持为用户查询提供新鲜的搜索结果，搜索引擎必须使那些陈旧的缓存实体无效。一个琐碎的失效机制是让索引器指出倒排索引更新的时机，从而让缓存使缓存里的所有查询失效。当索引经常更新时，缓存频繁的清空会严重影响它的命中率，这可能到达使缓存毫无用处的程度。

### 缓存更新技术

在缓存更新技术中，被预测为陈旧且不久将被用户查询的缓存内容通过搜索后台重新计算相关查询而得到更新。这类技术利用搜索引擎后台的闲置周期（如深夜）来重新计算被猜测为结果已陈旧的查询。这类技术不需要结果缓存与搜索系统后台交互来决定哪些查询要更新。

然而，不使用后台提供的索引更新的具体信息，仅利用一些历史查询记录和查询特征来辨别查询是否陈旧和预测查询何时将再次被用户发起是一个相当困难的任务。这导致许多查询结果被不必要地重新计算而未对缓存新鲜度产生正面影响。

Cambazoglu等人[3]提出一个新颖的算法，用生存时间值来设置缓存实体(cache entries)的过期时间并选择性地通过向搜索引擎后台发送更新查询来更新缓存结果。这个启发式算法结合了查询的频率和查询结果在缓存里的时间来给需要被更新的缓存实体排序。此外，通过设置查询更新率展示了一种将搜索引擎后台闲置周期考虑进去的机制。实际后台工作负载表明，这个算法不仅提高了缓存命中率，也减少了缓存结果的平均年龄。该算法的一个实现目前被yahoo所使用。

Jonassen等人[1]提出线下和线上两种预取策略来更新缓存中可能陈旧的查询结果。线下预取策略,从[17]的图2中得出许多查询会在上次提交后的24小时左右再次被发起。线上预取策略利用查询的频率，查询重复发起的时间间隔，查询词数等属性用机器学习的方法来预测查询再次被发起的时间。由此确定更新特定查询缓存结果的时机。

### 缓存内容失效技术

在缓存内容失效技术中，索引系统向结果缓存传递关于最近倒排索引的更新信息。这些信息被缓存用来辨别可能的陈旧查询结果。搜索结果缓存模块对缓存中的每个查询判断索引的变化是否会导致这个查询的结果陈旧，并将查询标记为失效，例如将它的结果视为未被缓存。

与缓存更新相比，缓存模块与索引系统的信息交互将占用更多的资源。

Blanco等人[2]提出一种缓存内容失效预测(Cache Invalidation Predictor CIP)方案。搜索引擎后台为增、删或修改的文档产生文档概要，并发送给缓存。缓存根据文档概要判断文档的修改是否引起缓存中某查询结果陈旧。Alici等人[8]提出一种较为简单易于搜索引擎实现的缓存内容失效方案中，在文档被增、删和修改时仅需要相应调整倒排记录表和文档的版本时间戳。但在同等假正率的情况下未能达到采用CIP的缓存所达到的内容新鲜度。Bai等人[4]提出了一个查询驱动的缓存内容失效框架，失效决定只在有查询被发起且查询结果在缓存中时发生。它将索引的更新信息保存在子索引上来为失效决定提供依据。

### 生存时间机制

在上述两类技术中，陈旧决策由启发式算法得到而不能产生完美的准确度。某些查询结果陈旧了，却可能没有被及时鉴别出来。这样的查询的陈旧结果可能在缓存中存在很长一段时间。上述两类技术依靠补充机制：生存时间(Time-to-Live TTL)，来补救这个问题。

Sazoglu[5]等人考虑三种设置TTL的方式，基于时间的TTL、基于频率的TTL和基于用户点击的TTL，并考虑这三种方式的析取和合取。通过实验表明基于时间的TTL和基于频率的TTL以析取的方式结合能达到最好的性能。

## 信息检索技术

搜索引擎的缓存技术经常要涉及文档评分、倒排索引压缩等信息检索技术。

Introduction to Information Retrieval[34]介绍了索引构建方法，索引压缩方法，文档评分方法和链接分析方法等。

在一份给定的文件里，词频 (term frequency, TF) 指的是某一个给定的词语在该文件中出现的次数。这个数字通常会被归一化（分子一般小于分母 区别于IDF），以防止它偏向长的文件。（同一个词语在长文件里可能会比短文件有更高的词频，而不管该词语重要与否。）

　　逆向文件频率 (inverse document frequency, IDF) 是一个词语普遍重要性的度量。某一特定词语的IDF，可以由总文件数目除以包含该词语之文件的数目，再将得到的商取对数得到。

　　某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的TF-IDF。因此，TF-IDF倾向于过滤掉常见的词语，保留重要的词语。

TFIDF的主要思想是：如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。TFIDF实际上是：TF \* IDF，TF词频(Term Frequency)，IDF反文档频率(Inverse Document Frequency)。TF表示词条在文档d中出现的频率（另一说：TF词频(Term Frequency)指的是某一个给定的词语在该文件中出现的次数）。IDF的主要思想是：如果包含词条t的文档越少，也就是n越小，IDF越大，则说明词条t具有很好的类别区分能力。如果某一类文档C中包含词条t的文档数为m，而其它类包含t的文档总数为k，显然所有包含t的文档数n=m+k，当m大的时候，n也大，按照IDF公式得到的IDF的值会小，就说明该词条t类别区分能力不强。（另一说：IDF反文档频率(Inverse Document Frequency)是指果包含词条的文档越少，IDF越大，则说明词条具有很好的类别区分能力。）但是实际上，如果一个词条在一个类的文档中频繁出现，则说明该词条能够很好代表这个类的文本的特征，这样的词条应该给它们赋予较高的权重，并选来作为该类文本的特征词以区别与其它类文档。这就是IDF的不足之处.在一份给定的文件里，词频（term frequency，TF）指的是某一个给定的词语在该文件中出现的频率。逆向文件频率（inverse document frequency，IDF）是一个词语普遍重要性的度量。某一特定词语的IDF，可以由总文件数目除以包含该词语之文件的数目，再将得到的商取对数得到：idf(t)=lgN/(df(t)).其中N是所有文档数目，df(t)是包含词项t的文档数目。

向量空间模型(VSM)把对文本内容的处理简化为[向量空间](http://baike.baidu.com/view/327493.htm)中的向量运算，并且它以空间上的相似度表达语义的相似度，直观易懂。当文档被表示为文档空间的向量，就可以通过计算向量之间的相似性如余弦相似度来度量文档间的相似性。

信息检索系统中的两个主要数据结构:词典及倒排索引。索引压缩就是对这两个数据结构的各种压缩技术,这些技术对于构建高效的 IR 系统非常关键。 进行压缩的一个优点显而易见:它能够节省磁盘空间。要达到 1∶4 的压缩比是非常容易的,也就是说可以降低 75%的索引存储开销。

索引压缩还有两个隐含的优点。第一是能增加高速缓存(caching)技术的利用率。在搜索 系统中,词典中某些条目及其索引往往比其他条目及其索引的使用更频繁。例如,如果将一个频 繁使用的查询词项 t 的倒排记录表放到高速缓存中,那么对仅由 t 构成的查询进行应答所需要的 计算完全可以在内存中完成。如果采用压缩技术,那么高速缓存中就可以放更多的信息。当查询词项 t 的信息放在高速缓存时,处理查询 t 便不再需要进行磁盘操作,而只需将其倒排记录表在内存中解压缩即可。因此,我们能充分减少 IR 系统的应答时间。由于内存比磁盘更贵,所以, 相对于磁盘空间的减少,采用高速缓存技术带来的速度提升是采用压缩技术的更主要的原因。

第二个隐含的优点是,压缩能够加快数据从磁盘到内存的传输速度。高效的解压缩算法在现代硬件上运行相当快,这样将压缩的数据块传输到内存并解压所需要的总时间往往会比将未压缩的数据块传输到内存要快。举例来说,即使会增加在内存进行解压缩的开销,我们也可以 通过加载一个小很多的压缩倒排记录表来减少 I/O 时间。因此,在大部分情况下,使用压缩倒排记录表的检索系统会比没用压缩的系统的运行速度要快。

如果压缩的主要目的是为了节省磁盘空间,那么压缩算法的速度就不用特别考虑。但是, 如果要提高高速缓存利用率和磁盘到内存的传输率,则解压缩的速度必须要快。本章介绍的压缩算法都非常高效,都可以达到上面提到的索引压缩的全部 3 个目标。

采用定长方法来存储词项存在着明显的空间浪费。 一种解决上述缺陷的方法是,将所有的词项存成一个长字符串，并给每个词项增加一个定位指针,它在指向下一词项的指针同时也标识着 当前词项的结束。

想象一下在文档 集中遍历文档来寻找某个高频词项(如 computer)的过程:我们会找到一篇包含 computer 的文 档,然后可能会跳过几篇不包含它的文档,之后又会找到另一篇包含 computer 的文档。这个过 程可以不断循环下去。这里面最关键的思路就是倒排记录表 中文档 ID 之间的间距(gap)不大,因此可以考虑用比 20 比特短很多的位数来表示它。实际上, 对于一些高频词(如 the 和 and)来说,绝大部分间距都是 1。当然,对于只在文档集中出现一 两次的罕见词,其间距的数量级和文档 ID 的数目是一样的,因 此仍然需要 20 比特。为了对这种间距分布的情况进行空间压缩,需要使用一种变长编码方法, 它可以对短间距采用更短的位数来表示。

VB(Variable byte, 可变字节)编码利用整数个字节来对间距编码。字节的后 7 位是间距的有效编码区,而第 1 位是延续位(continuation bit)。如果该位为 1,则表明本字节是某个间距编 码的最后一个字节,否则不是。要对一个可变字节编码进行解码,可以读入一段字节序列,其中前面的字节的延续位都为 0,而最后一个字节的延续位为 1。根据上述标识可以把每个字节的 7 位部分抽取出来并连接在一起形成编码。

链接分析方法主要有PageRank和HITS。PageRank是Google用于用来标识网页的等级/重要性的一种方法，是Google用来衡量一个网站的好坏的唯一标准。在揉合了诸如Title标识和Keywords标识等所有其它因素之后，Google通过PageRank来调整结果，使那些更具“等级/重要性”的网页在搜索结果中另网站排名获得提升，从而提高搜索结果的相关性和质量。PageRank值是相应的马尔可夫链的稳态概率。PageRank的计算充分利用了两个假设：数量假设和质量假设。步骤如下：(1)在初始阶段：网页通过链接关系构建起Web图，每个页面设置相同的PageRank值，通过若干轮的计算，会得到每个页面所获得的最终PageRank值。随着每一轮的计算进行，网页当前的PageRank值会不断得到更新。(2)在一轮中更新页面PageRank得分的计算方法：在一轮更新页面PageRank得分的计算中，每个页面将其当前的PageRank值平均分配到本页面包含的出链上，这样每个链接即获得了相应的权值。而每个页面将所有指向本页面的入链所传入的权值求和，即可得到新的PageRank得分。当每个页面都获得了更新后的PageRank值，就完成了一轮PageRank计算。

HITS 算法是由康奈尔大学( Cornell University ) 的JonKleinberg 博士于1998 年首先提出的,为IBM 公司阿尔马登研究中心( IBM Almaden Research Center) 的名为“CLEVER”的研究项目中的一部分。一个网页重要性的分析的算法，根据一个网页的入度（指向此网页的超链接）和出度（从此网页指向别的网页）来衡量网页的重要性。其最直观的意义是如果一个网页的重要性很高，则他所指向的网页的重要性也高。一个重要的网页被另一个网页所指，则表明指向它的网页重要性也会高。指向别的网页定义为Hub值,被指向定义为Authority值。通常HITS算法是作用在一定主题的，比如一个以程序开发为主题网页，指向另一个以程序开发为主题的网页，则另一个网页的重要性就可能比较高，但是指向另一个购物类的网页则不一定。在限定主题之后根据网页的出度和入度建立一个矩阵，通过矩阵的迭代运算和定义收敛的阈值不断对两个向量Authority和Hub值进行更新直至收敛

# 结论

本文从搜索引擎缓存技术背景开始，分别总结了搜索引擎缓存技术应用背景，面临的挑战，概述了搜索引擎缓存技术的研究方向和发展趋势，重点总结了面向无限容量缓存和持续更新索引的缓存技术，本文所处的研究项目就是使用搜索引擎缓存技术来达到搜索引擎效率优化的目的，实现资源的有效利用和提高搜索引擎用户的满意度。

使用缓存技术的搜索引擎环境有和其他环境的共同特征，但是也有不同之处。对于这样的环境，缓存的实体体现出不同的特征，如何组织和管理这些存储资源，以达到有效的利用是有必要研究和实现的任务。本文总结了在不同的环境在搜索引擎缓存技术使用的方法，这些方法在其应用场景下有一定的合理性，但是都存在着一些限制和问题。

总之，面向实时检索的搜索引擎缓存技术需要更为深入的研究。

# 主要参考文献

1. S.Jonassen, B.B. Cambazoglu, and F. Silvestri. Prefetching query results and its impact on search engines. *In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. SIGIR’12.ACM.*
2. [Roi Blanco](http://dblp.uni-trier.de/pers/hd/b/Blanco:Roi), [Edward Bortnikov](http://dblp.uni-trier.de/pers/hd/b/Bortnikov:Edward), [Flavio Junqueira](http://dblp.uni-trier.de/pers/hd/j/Junqueira:Flavio), [Ronny Lempel](http://dblp.uni-trier.de/pers/hd/l/Lempel:Ronny), [Luca Telloli](http://dblp.uni-trier.de/pers/hd/t/Telloli:Luca), [Hugo Zaragoza](http://dblp.uni-trier.de/pers/hd/z/Zaragoza:Hugo). Caching search engine results over incremental indices. [SIGIR 2010](http://dblp.uni-trier.de/db/conf/sigir/sigir2010.html#BlancoBJLTZ10): 82-89
3. [Berkant Barla Cambazoglu](http://dblp.uni-trier.de/pers/hd/c/Cambazoglu:Berkant_Barla), [Flavio Paiva Junqueira](http://dblp.uni-trier.de/pers/hd/j/Junqueira:Flavio_Paiva), [Vassilis Plachouras](http://dblp.uni-trier.de/pers/hd/p/Plachouras:Vassilis), [Scott A. Banachowski](http://dblp.uni-trier.de/pers/hd/b/Banachowski:Scott_A=), [Baoqiu Cui](http://dblp.uni-trier.de/pers/hd/c/Cui:Baoqiu), [Swee Lim](http://dblp.uni-trier.de/pers/hd/l/Lim:Swee),[Bill Bridge](http://dblp.uni-trier.de/pers/hd/b/Bridge:Bill). A refreshing perspective of search engine caching. [WWW 2010](http://dblp.uni-trier.de/db/conf/www/www2010.html#CambazogluJPBCLB10): 181-190
4. [Xiao Bai](http://dblp.uni-trier.de/pers/hd/b/Bai_0002:Xiao), [Flavio Paiva Junqueira](http://dblp.uni-trier.de/pers/hd/j/Junqueira:Flavio_Paiva). Online result cache invalidation for real-time web search. [SIGIR 2012](http://dblp.uni-trier.de/db/conf/sigir/sigir2012.html#BaiJ12): 641-650
5. [Fethi Burak Sazoglu](http://dblp.uni-trier.de/pers/hd/s/Sazoglu:Fethi_Burak), [Berkant Barla Cambazoglu](http://dblp.uni-trier.de/pers/hd/c/Cambazoglu:Berkant_Barla), [Rifat Ozcan](http://dblp.uni-trier.de/pers/hd/o/Ozcan:Rifat), [Ismail Sengör Altingövde](http://dblp.uni-trier.de/pers/hd/a/Alting=ouml=vde:Ismail_Seng=ouml=r), [Özgür Ulusoy](http://dblp.uni-trier.de/pers/hd/u/Ulusoy:=Ouml=zg=uuml=r). Strategies for setting time-to-live values in result caches. [CIKM 2013](http://dblp.uni-trier.de/db/conf/cikm/cikm2013.html#SazogluCOAU13): 1881-1884
6. [Ricardo A. Baeza-Yates](http://dblp.uni-trier.de/pers/hd/b/Baeza=Yates:Ricardo_A=), [Flavio Junqueira](http://dblp.uni-trier.de/pers/hd/j/Junqueira:Flavio), [Vassilis Plachouras](http://dblp.uni-trier.de/pers/hd/p/Plachouras:Vassilis), [Hans Friedrich Witschel](http://dblp.uni-trier.de/pers/hd/w/Witschel:Hans_Friedrich). Admission Policies for Caches of Search Engine Results. [SPIRE 2007](http://dblp.uni-trier.de/db/conf/spire/spire2007.html#Baeza-YatesJPW07): 74-85
7. [Hongyuan Ma](http://dblp.uni-trier.de/pers/hd/m/Ma:Hongyuan), [Wei Liu](http://dblp.uni-trier.de/pers/hd/l/Liu:Wei), [Bingjie Wei](http://dblp.uni-trier.de/pers/hd/w/Wei:Bingjie), [Liang Shi](http://dblp.uni-trier.de/pers/hd/s/Shi:Liang), [Xiuguo Bao](http://dblp.uni-trier.de/pers/hd/b/Bao:Xiuguo), [Lihong Wang](http://dblp.uni-trier.de/pers/hd/w/Wang:Lihong), [Bin Wang](http://dblp.uni-trier.de/pers/hd/w/Wang:Bin). PAAP: prefetch-aware admission policies for query results cache in web search engines. [SIGIR 2014](http://dblp.uni-trier.de/db/conf/sigir/sigir2014.html#MaLWSBWW14): 983-986
8. [Sadiye Alici](http://dblp.uni-trier.de/pers/hd/a/Alici:Sadiye), [Ismail Sengör Altingövde](http://dblp.uni-trier.de/pers/hd/a/Alting=ouml=vde:Ismail_Seng=ouml=r), [Rifat Ozcan](http://dblp.uni-trier.de/pers/hd/o/Ozcan:Rifat), [Berkant Barla Cambazoglu](http://dblp.uni-trier.de/pers/hd/c/Cambazoglu:Berkant_Barla), [Özgür Ulusoy](http://dblp.uni-trier.de/pers/hd/u/Ulusoy:=Ouml=zg=uuml=r). Timestamp-based result cache invalidation for web search engines. [SIGIR 2011](http://dblp.uni-trier.de/db/conf/sigir/sigir2011.html#AliciAOCU11): 973-982
9. [Xiao Bai](http://dblp.uni-trier.de/pers/hd/b/Bai_0002:Xiao), [Flavio Paiva Junqueira](http://dblp.uni-trier.de/pers/hd/j/Junqueira:Flavio_Paiva), [Adam Silberstein](http://dblp.uni-trier.de/pers/hd/s/Silberstein:Adam). Cache refreshing for online social news feeds. [CIKM 2013](http://dblp.uni-trier.de/db/conf/cikm/cikm2013.html#BaiJS13): 787-792
10. Mauricio Marin, Veronica Gil-Costa, Carlos Gomez-Pantoja. New caching techniques for web search engines. In: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, 2010
11. R. Lempel, and S. Moran. Predictive caching and prefetching of query results in search engines. In WWW, pp 19-28, 2003
12. E. Markatos. On caching search engine query results. Computer Communications, 24(7), 2000.
13. T. Fagni, R. Perego, F. Silvestri, and S. Orlando. Boosting the performance of Web search engines. Caching and prefetching query results by exploiting historical usage data. ACM TOIS, 24(1):51-78, 2006.
14. R. Baeza-Yates, A. Gionis, F. Junqueira, V. Murdock, V. Plachouras, and F.Silvestri. Design trade-offs for search engine caching. ACMTWEB, 2(4):1-28,2008.
15. D. Puppin, F. Silvestri, R. Perego, and R. Baeza-Yates. Load-balancing and caching for collection selection architectures. In INFOSCALE, 2007.
16. D. Puppin, F. Silvestri, R. Perego, and R. Baeza-Yates. Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load. To appear in TOIS, 2009.
17. B.B. Cambazoglu and R. Baeza-Yates. Scalability challenges in web search engines. In M. Melucci, R. Baeza-Yates, and W. B. Croft, editors, Advanced Topics in Information Retrieval, volume 33 of The Information Retrieval Series, pages 27-50. Springer Berlin Heidelberg,2011.
18. L. A. Barroso, J.Dean, and U. Holzle. Web search for a planet: the Google cluster architecture. IEEE Micro,23(2):22-28,2003
19. 马宏远,王斌.一种基于预取感知接纳策略的查询结果缓存方法[J]计算机研究与发展，2012(S1)
20. 马宏远,王斌.一种基于查询特性的查询结果缓存与预取方法[J]中文信息学报,2011(5):37-43
21. 马宏远,王斌.基于日志分析的搜索引擎查询结果缓存研究[J]计算机研究与发展,2012
22. China internet Network Information Center (CNNIC). The 34th Statistical Report on Internet Development in China. [2014-07-21] http: //www.cnnic.net. cn/hlwfzyj/hlwxzbg/hlwtjbg/201407/P020140721507223212132. pdf
23. Cockburn A, Jones S. Which way now? Analysing and easing inadequacies in WWW navigation. International Journal of Human-Computer Studies, 1996, 45:105-129
24. Tauscher L, Greenberg S. How people revisit web pages: Empirical findings and implications for the design of history system. International Journal of Human-Computer Studies, 1997,47:97-137
25. Silverstein C, Henzinger M, Marais H,et al. Analysis of a very large web search engine query log. SIGIR 1998:6-12
26. Baeze-Yates R. Web Usage Mining in Search Engines. Reading, MA: Addison Wesley, 1999:307-322
27. Baeze-Yates R. Gionis A, Junquerira F, et al. The impact of caching on search engines. SIGIR 2007:183-190
28. Zhang J, Long X, Suel T, et al. Performance of compressed inverted list caching in search engines. WWW 2008:387-396
29. Xie Y, O’Hallaron D R. Locality in search engine queryies and its implications for caching. IEEE 2002:1238-1247
30. Li Y, Zhang S, Wang B, et at. Characteristics of Chinese Web searching: A large-scale analysis of Chinese query logs. Journal of Computational Information Systems, 2008, 4(3):1127-1136
31. [Xiaohui Long](http://dblp.uni-trier.de/pers/hd/l/Long:Xiaohui), [Torsten Suel](http://dblp.uni-trier.de/pers/hd/s/Suel:Torsten). Three-Level Caching for Efficient Query Processing in Large Web Search Engines. [World Wide Web 9(4)](http://dblp.uni-trier.de/db/journals/www/www9.html#LongS06): 369-395 (2006)
32. [Patricia Correia Saraiva](http://dblp.uni-trier.de/pers/hd/s/Saraiva:Patricia_Correia), [Edleno Silva de Moura](http://dblp.uni-trier.de/pers/hd/m/Moura:Edleno_Silva_de), [Rodrigo C. Fonseca](http://dblp.uni-trier.de/pers/hd/f/Fonseca:Rodrigo_C=), [Wagner Meira Jr.](http://dblp.uni-trier.de/pers/hd/m/Meira_Jr=:Wagner), [Berthier A. Ribeiro-Neto](http://dblp.uni-trier.de/pers/hd/r/Ribeiro=Neto:Berthier_A=), [Nivio Ziviani](http://dblp.uni-trier.de/pers/hd/z/Ziviani:Nivio). Rank-Preserving Two-Level Caching for Scalable Search Engines. [SIGIR 2001](http://dblp.uni-trier.de/db/conf/sigir/sigir2001.html#SaraivaMFMRZ01): 51-58
33. [Gleb Skobeltsyn](http://dblp.uni-trier.de/pers/hd/s/Skobeltsyn:Gleb), [Flavio Junqueira](http://dblp.uni-trier.de/pers/hd/j/Junqueira:Flavio), [Vassilis Plachouras](http://dblp.uni-trier.de/pers/hd/p/Plachouras:Vassilis), [Ricardo A. Baeza-Yates](http://dblp.uni-trier.de/pers/hd/b/Baeza=Yates:Ricardo_A=). ResIn: a combination of results caching and index pruning for high-performance web search engines. [SIGIR 2008](http://dblp.uni-trier.de/db/conf/sigir/sigir2008.html#SkobeltsynJPB08): 131-138

[34]Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze. Introduction to Information Retrieval[M]. Cambridge University Press. 2008.