

摘要

P2P 搜索引擎是一种基于 P2P 网络结构的全文检索引擎，它是网络技术研究中的一个新兴领域，现有的 P2P 搜索技术在如何面对大规模的网络节点保持系统的稳定性、可靠性，减少对超级节点的依赖及对网络节点共享资源的发现、索引、复杂搜索、深度搜索等方面还存在一些不足，本文在如何实现网络稳定可靠、对超级节点依赖少及高效的对资源进行发现索引、支持复杂搜索、中文处理和深度搜索等方面进行了深入的研究。

针对传统中文分词算法的缺陷，本文提出了一种基于 X-树结构的中文分词算法 (CSECHSplitor)，CSECHSplitor 算法首先在 X-树结构初始节点查找首字哈希值，再根据首字孩子节点指针在基于有序表的孩子节点上查找后继元素，通过对比基于哈希表的新词词库，动态移动首字指针，解析出数据中有效的分词单位。在 CSECHSplitor 基础上本文设计并实现了配套的全文检索系统；针对传统 P2P 搜索技术所采用的网络结构不足，提出了一种改进的三层半分布混合式 P2P 网络结构，该网络结构将系统网络节点按功能划分为引导节点、超级节点、普通节点，并扩展引导节点与超级节点定义——采用纯 P2P 模式连接的子网络，尽最大可能的避免传统模式下由于超级节点失效而导致局部网络失效的问题，结合三层混合式半分布 P2P 网络结构与全文检索系统的基础上实现了一种基于 P2P 的中文搜索引擎 P2P Chinese Search Engine (P2PCSE)。

通过构建了一个基于局域网的实验环境，对 P2PCSE 系统中文分词算法的效率、准确度，P2PCSE 系统网络的可靠性、可扩展性及 P2PCSE 系统资源搜索的覆盖深度、准确度、满意度进行了详细的测试。实验结果表明 P2PCSE 系统具有网络稳定可靠、资源覆盖广，支持复杂搜索的优点。

关键字:对等网络, 中文分词, 搜索引擎

Abstract

For overcoming the disadvantages in the traditional Chinese word segmentation algorithm, a X- tree based Chinese word segmentation algorithm (CSECHSplitor) is proposed in this paper. It firstly looks up the hash value of the first word in the initial node of the X- tree, then subsequent elements in child node based on ordered list according to pointer of the first word's child node. By comparing the new words' dictionary based on the hash table and dynamically moving the pointer of the first word, effective word segmentation can be parsed. A compatible full-text retrieval system is also implemented based on CSECHSplitor. To improve network structure of the traditional P2P searching technology, this paper also proposed an improved distribution of the three semi-hybrid P2P network (IDOTSP2PNETWORK), which defines the network nodes as guiding nodes, super nodes and normal nodes according to their functions. The guiding nodes and super nodes are organized as a sub-network with pure P2P model. In the foundation of IDOTSP2PNETWORK and full-text searching system, a P2P based P2P Chinese searching engine (P2PCSE) is set up.

By building a lan-based experimental environment, tests for evaluating P2PCSE's efficiency, accuracy, reliability, scalability and the depth coverage, satisfaction in its resource searching are conducted. The results show that the P2PCSE system is stable and reliable in networking, and also supports complex searching.

Keywords: P2P, Chinese Word Segmentation, Search Engine

南京邮电大学学位论文原创性声明

本人声明所提交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南京邮电大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

研究生签名： 刘华 日期： 2009.04.08

南京邮电大学学位论文使用授权声明

南京邮电大学、中国科学技术信息研究所、国家图书馆有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其它复制手段保存论文。本文电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权南京邮电大学研究生部办理。

研究生签名： 刘华 导师签名： 王华 日期： 2009.04.08

第一章 引言

1.1 论文的选题来源及其研究意义

本文基于华为基金项目—P2P 中文搜索引擎,项目从中文搜索引擎[1]、P2P[2~4]网络结构两个方面详细研究了 P2P 中文搜索引擎技术。

随着现代计算机软硬件技术的发展, P2P 网络 (Peer-to-Peer Network) (也称计算机对等互联网)得到前所未有的快速发展,据统计 P2P 流量已经超过传统的 WEB 流量,成为因特网流量的主要消费者。

在 P2P 网络中由于不存在绝对的中心服务器,每个节点既是服务器,又是客户机。一方面为其他节点提供各种数据服务,同时有享用其他节点提供的服务,网络中的节点地位是对等的。各节点统一构成负载均衡的 P2P 网络。节点越多,系统负载能力越强,整个系统的可扩展性就越强。

对于现有的搜索引擎而言, P2P 技术不仅是简单的节约了大规模的服务器成本和提升了搜索速度,更重要的是 P2P 技术大大增加了传统搜索引擎的数据覆盖范围,促成了搜索模式的多元化,权利,责任,义务的合理并行化。将 P2P 技术与搜索结合一起的 P2P 搜索技术[5, 6],解决了当前互联网分散备份和无备份,分散索引和无索引的隐患,将搜索的格局由统而死转变成了分而活,多而散,发动了全网力量参与搜索的新方案。

因此 P2P 技术融合中文搜索技术而构成的 P2P 中文搜索引擎的研究与应用对现有的搜索技术是一个重大意义的提升。

1.2 本文研究的主要内容及创新之处

在深入研究 P2P 技术和中文搜索引擎技术的基础上,融合搜索引擎中的全文检索[7]、P2P 网络结构,提出了基于 P2P 的中文搜索引擎 P2P Chinese Search Engine (P2PCSE)解决方案。

本文的主要研究内容及创新之处包含:

- 在介绍中文搜索引擎技术的基础上着重对中文搜索引擎中的关键技术——中文分词做了重点分析,总结概括了目前中文分词技术的一般方法及存在的缺陷。

- 介绍 P2P 搜索技术, 对比 P2P 搜索与 C/S 模式的优缺点, 并总结归纳了常见 P2P 搜索技术采用的网络结构的优缺点。
- 在相关技术研究的基础上对 P2PCSE 系统需求进行了分析, 提出了一种基于 X-树结构的中文分词算法及改进三层半分布式网络结构, 在两者结合的基础上, 对系统进行总体设计, 对系统功能模块进行了细致的划分, 详细阐述了 P2PCSE 系统运行流程及对实现 P2PCSE 网络结构所使用的主要协议进行了设计和实现。
- 对 P2PCSE 中的关键部分——全文检索、网络结构进行了一一探讨, 对全文检索子系统进行了详细的模型分析, 按模块划分为分词处理模块、文件处理模块、文件索引储存查询模块, 分别对其进行详细分析及实现; 同时对网络结构模型进行了介绍, 对关键部分如超级节点的产生、新节点的加入等进行了一一实现。
- 最后实现了 P2PCSE 原型系统。并对系统搜索搜索流程的实现、用户界面进行了一一介绍。对全文检索系统中的中文分词算法进行对比测试, 对全文检索子系统的满意度和准确度进行了测试, 通过新节点并发加入, 节点的失效及退出、索引数据同步、文件搜索请求等实验对 P2PCSE 系统使用的网络结构各方面的性能进行了测试。通过上述实验的结果得出 P2PCSE 系统可靠、稳定、可扩展, 对复杂查询良好的支持及 P2PCSE 系统资源搜索结果的准确度和满意度均较高。

1.3 论文结构安排

本文完成了 P2PCSE 系统的设计和实现, 并对系统各方面的功能进行了详细的测试与分析。本文共分七大章节, 组织如下:

第一章 说明了论文的选题来源及其研究意义, 并对本文研究的主要内容及创新之处做了介绍。

第二章 首先对中文搜索引擎技术的发展状况做了介绍, 并对其技术结构尤其是其中的关键技术——中文分词技术做了重点剖析。接着对 P2P 搜索技术做了简单介绍, 对比得出 P2P 模式与 C/S 模式的优缺点。最后按照 P2P 网络结构对常见 P2P 搜索技术进行了详细的分析, 并指出目前技术所采用网络结构的缺点。

第三章 提出了 P2PCSE 系统的总体设计, 对其设计目标、功能模块、流程结构、通讯协议、关键技术进行了详细的定义及分析。

第四章 对 P2PCSE 系统的关键部分——全文索引子系统进行了详细的分析和实现, 特别是对其中由本文提出的基于 X-树结构的中文分词算法进行了定义分析和实现。

第五章 对 P2PCSE 系统的关键部分——网络结构进行了详细的模型定义和说明，并对实现网络结构的关键协议进行了设计和实现。

第六章 实现了 P2PCSE 原型系统，对系统搜索流程做了详细介绍，对其界面进行了详细的介绍，对全文检索子系统基于 X-树结构的中文分词算法进行了测试，对全文检索子系统的准确度和满意度进行了详细测试，对 P2PCSE 系统网络结构在各种并发请求下分别进行测试。

第七章 总结本文所做的工作，并指出 P2PCSE 系统有待进一步完善之处和未来发展的方向。

第二章 相关技术研究

本章主要对 P2PCSE 核心相关的中文搜索引擎技术和 P2P 搜索技术所采用的网路结构进行了详细分析。对中文搜索引擎的发展概述、中文搜索引擎的基本技术特别是对影响搜索结果的中文分词技术进行了详细的分析并指出常见中文分词技术的缺陷。在介绍 P2P 搜索技术的基础上对比 P2P 模式与 C/S 模式的优缺点,并对常见 P2P 搜索技术的网络结构进行了总结概括,指出目前 P2P 搜索技术所采用网络结构的缺点。

2.1 中文搜索引擎技术

2.1.1 中文搜索引擎技术概述

互联网在近 10 年飞速发展,互联网已经深入人们的生活,并慢慢改变人们的生活,从“网络广告”到“拇指经济”,从“网络游戏”到“搜索力经济”。目前搜索引擎已经成为互联网行业中最受人们关注的焦点。

搜索引擎的基础技术是全文检索,20 世纪 60 年代,国外就已经开始对全文检索技术 [7] 进行研究。其核心是以文本信息的索引和检索,一般用于企事业单位。随着互联网的发展,搜索引擎在全文检索技术上发展起来,并得到了广泛的应用。

搜索引擎结合互联网发展的特点形成了三种典型的类型 [9]:

- 全文检索搜索引擎

国外具代表性的有 Google、yahoo、AllTheWeb 等,国内著名的有百度。它们都是通过从互联网上提取的各个网站的信息(以网页文字为主)而建立的数据库,检索与用户查询条件匹配的相关记录,然后按一定的排列顺序将结果返回给用户,也是目前常规意义上的搜索引擎。

- 目录搜索引擎

目录索引虽然有搜索功能,但在严格意义上算不上是真正的搜索引擎,仅仅是按目录分类的网站链接列表而已。用户完全可以不用进行关键词查询,仅靠分类目录也可找到需要的信息。国外比较著名的目录索引搜索引擎有 Yahoo、Open Directory Project、LookSmart 等。国内的搜狐、新浪、网易搜索也都具有这一

类功能。

- 元搜索引擎

元搜索引擎在接受用户查询请求时，同时在其它多个引擎上进行搜索，并将结果返回给用户。著名的元搜索引擎有 Dogpile、Vivisimo 等。在搜索结果排列方面，有的直接按来源引擎排列搜索结果，如 Dogpile，有的则按自定的规则将结果重新排列组合，如 Vivisimo。

其他的像新浪、网易、A9 等搜索引擎都是调用其它全文检索搜索引擎，或者在其搜索结果的基础上做了二次开发。

2.1.2 中文搜索引擎基本技术分析

搜索引擎的门槛主要是技术门槛，包括网页数据的快速采集、海量数据的索引和存储、搜索结果的相关性排序、搜索效率的毫秒级要求、分布式处理和负载均衡、自然语言的理解技术等等，这些都是搜索引擎的门槛。

搜索引擎可以主要划分为如下几大基础技术模块[10]：

1) 系统架构

同所有软件项目一样，搜索引擎的系统架构在整个系统中起中很关键的作用。分析搜索引擎实现原理我们可以细分为四步：

Step1) 抓取互联网数据

通过自动抓取网页的蜘蛛程序广度遍历整个互联网，并且保存访问过的网站页面，以供程序分析。

Step2) 建立索引数据库

由索引系统对收集到的网页进行分析，提取相关网页信息，根据一定的相关度算法进行大量复杂计算，得到每一个网页针对页面内容中及超链中每一个关键词的相关度，然后用这些相关信息建立网页索引数据库。

Step3) 搜索索引数据库

当用户输入关键词搜索后，分解搜索请求，由搜索系统程序从网页索引数据库中找到符合该关键词的所有相关网页。

Step4) 搜索结果的排序

所有相关网页针对该关键词的相关信息在索引库中都有记录，只需综合相关信息和网页级别形成相关度数值，然后进行排序，相关度越高，排名越靠前。最后由页面生成

系统将搜索结果的链接地址和页面内容摘要等内容组织起来返回给用户。

2) 网络蜘蛛

网络蜘蛛(Web Spider), 是通过网页的链接地址来寻找网页, 从网站某一个页面(通常是首页)开始, 读取网页的内容, 找到在网页中的其它链接地址, 然后通过这些链接地址寻找下一个网页, 这样一直循环下去, 直到把这个网站所有的网页都抓取完为止。如果把整个互联网当成一个网站, 那么网络蜘蛛就可以用这个原理把互联网上所有的网页都抓取下来。

在抓取网页的时候, 网络蜘蛛一般有两种策略: 广度优先和深度优先。广度优先是指网络蜘蛛会先抓取起始网页中链接的所有网页, 然后再选择其中的一个链接网页, 继续抓取在此网页中链接的所有网页。这是最常用的方式, 因为这个方法可以让网络蜘蛛并行处理, 提高其抓取速度。深度优先是指网络蜘蛛会从起始页开始, 一个链接一个链接跟踪下去, 处理完这条线路之后再转入下一个起始页, 继续跟踪链接。

3) 排序技术

在搜索引擎的数据库中, 普通一个搜索往往有上百万条结果, 如何按照某种方式对结果排序最终展示给访问者, 是搜索引擎中一个基本而关键的技术。排序技术的好坏直接影响到搜索引擎的性能优劣。经过多年的发展, 目前已经有多种排序技术: PageRank[11][12]技术、超链分析技术[13]、HillTop 技术[14]、锚文本[15]、页面版式。

2.1.3 中文搜索引擎关键技术——中文分词

我们知道中文和英文的书写方式是不一样的, 英文是以词为单位的, 而中文是以字为单位的。英文中的词是以空格为间隔的, 而中文必须把句子中的字连接起来才能正确理解意思, 例如英文句子: I come from Beijing., 用中文表示就是“我来自北京”, 计算机很容易通过空格分隔知道, Beijing 是一个单词, 但是对于中文, 必须把“北”和“京”连接起来才能正确理解“北京”的意思, 在中文搜索引擎中把中文汉字序列切成有意义的序列就叫中文分词。在搜索引擎中, 最重要的并不是找到所有结果, 因为在上百亿的网页中找到结果并没有多大实际意义, 最重要的是把最相关的结果排在最上面, 这也就是相关排序技术, 而中文分词技术的准确程度, 直接影响了搜索结果的相关排序结果[15][16]。

2.1.3.1 词与分词单位

在各种语言中词是最小的能够独立活动的有意义的语言成分。是信息系统处理信息的

基本单位。词界 (word Boundary) 是词语之间的间隙。通过词界我们可以把一句话分解成多个独立的有意义的词。1992 年国家技术监督局颁布了《信息处理用现代汉语分词规范》在其中将“分词单位”明确定义为“汉语言信息处理用、具有确定的语义或语法功能的基本单位；包括不规范的规则限定的词和词组”，并给出了一些相关的规则来判断分词。该规范成为很多分词系统确定词的基础。

2.1.3.2 中文分词技术概括与分类

现有的分词算法可分为三大类：基于字符串匹配的分词方法、基于理解的分词方法和基于统计的分词方法[8][16~22]。

1) 基于字符串匹配的分词方法

1、基于字符串匹配的分词方法

这种方法又叫做机械分词方法，它是按照一定的策略将待分析的汉字串与一个“充分大的”机器词典中的词条进行配，若在词典中找到某个字符串，则匹配成功（识别出一个词）。按照扫描方向的不同，串匹配分词方法可以分为正向匹配和逆向匹配；按照不同长度优先匹配的情况，可以分为最大（最长）匹配和最小（最短）匹配；按照是否与词性标注过程相结合，又可以分为单纯分词方法和分词与标注相结合的一体化方法。常用的几种机械分词方法如下：

- a) 正向最大匹配法（由左到右的方向）；
- b) 逆向最大匹配法（由右到左的方向）；
- c) 最少切分（使每一句中切出的词数最小）。

还可以将上述各种方法相互组合，例如，可以将正向最大匹配方法和逆向最大匹配方法结合起来构成双向匹配法。由于汉语单字成词的特点，正向最小匹配和逆向最小匹配一般很少使用。一般说来，逆向匹配的切分精度略高于正向匹配，遇到的歧义现象也较少。统计结果表明，单纯使用正向最大匹配的错误率为 1/169，单纯使用逆向最大匹配的错误率为 1/245。但这种精度还远远不能满足实际的需要。实际使用的分词系统，都是把机械分词作为一种初分手段，还需通过利用各种其它的语言信息来进一步提高切分的准确率。

2、机械分词基本模型

对于机械分词可以建立起一般的格式化模型。其中典型的代表是 ASM 分词模型和基于图论的分词模型。ASM 分词模型提出较早，目前已经被普遍的认同和使用，后者提出时间教晚，它的核心是将图论思想引入分词中。

a) ASM(Automatic Segmentation Model)分词模型表示为 $ASM(d, a, m)$ ，其中 d 表示匹配方向，+1 表示正向，-1 表示逆向；每次匹配失败后增加/减少字符串长度，+1 代表增加一个字，-1 表示减少一个字。

m 代表最大/最小匹配标志，+1 为最大匹配，-1 为最小匹配。

b) 基于图论的分词模型是图论中的有向图的理论，将分词问题归结为图论中的最短路径问题。汉字字符串为 $S_0, S_1, S_2, \dots, S_N$ ；间隔位置为 $P_0, P_1, P_2, \dots, P_N$ ；构造有向无环图 $DAG = \{V, E, W\}$ ，定义为：

边：从 P_i 到 P_j 的边 $\langle P_i, P_j \rangle$ ；

顶点集合： $V = \{P_i, 10 \leq i \leq (n+1)\}$ ；

边的集合： $E = \{E_{ij} | 0 \leq i \leq n, 0 \leq j \leq (n+1), i < j\}$

边和顶点关系的集合： $W = \{E_{ij} = \langle P_i, P_j \rangle | C_1, \dots, C_{j-i} \text{ 为一个切分单位}\}$ 分词的过程，即可以表示为 DAG 中，从 P_0 到 P_{n+1} 的路径中利用评价函数选取最佳路径的过程。

2) 基于统计的分词方法

a) 基于统计的分词方法

从形式上看，词是稳定的字的组合，因此在上下文中，相邻的字同时出现的次数越多，就越有可能构成一个词。因此字与字相邻共现的频率或概率能够较好的反映成词的可信度。可以对语料中相邻共现的各个字的组合的频度进行统计，计算它们的互现信息。定义两个字的互现信息，计算两个汉字 X、Y 的相邻共现概率。互现信息体现了汉字之间结合关系的紧密程度。当紧密程度高于某一个阈值时，便可认为此字组可能构成了一个词。这种方法只需对语料中的字组频度进行统计，不需要切分词典，因而又叫做无词典分词法或统计取词方法[19]。

b) 基于统计分词的模型

设 $S = S_1, S_2, \dots, S_n$ 是待分字符串， $W = W_1, W_2, \dots, W_n$ 是切分结果， $P(W|S)$ 是字符串 S 切分为 W 的概率， W_1, W_2, \dots, W_n 是 S 的可能切分方案。基于统计的分词模型估计概率最大的分词方案 W' ，评价函数为

$$P(W'|S) = \max(P(W_i | S)) \quad i=1 \dots k$$

利用贝叶斯公式展开

$$P(W|S) = P(W) * P(S|W) / P(S)$$

$P(S)$ 为常数, $P(S|W)$ 为给定的词串 S 中, W 方案中出现 S 的概率 $P(W|S)=1$ 。因此, 该分词模型的问题实际上就是如何评估 $P(W)$ 的问题。 $P(W)$ 的展开一般采用马尔可夫 N 元语言模型。

c) 基于统计分词的缺陷

基于统计分词的方法也有一定的局限性, 会经常抽出一些共现频度高、但并不是词的常用字组, 例如“这一”、“之一”、“有的”、“我的”、“许多的”等, 并且对常用词的识别精度差, 时空开销大。实际应用的统计分词系统都要使用一部基本的分词词典(常用词词典)进行串匹配分词, 同时使用统计方法识别一些新的词, 即将串频统计和串匹配结合起来, 既发挥匹配分词切分速度快、效率高的特点, 又利用了无词典分词结合上下文识别生词、自动消除歧义的优点。

3) 基于理解的分词方法

这种分词方法是通过让计算机模拟人对句子的理解, 达到识别词的效果。其基本思想就是在分词的同时进行句法、语义分析, 利用句法信息和语义信息来处理歧义现象。它通常包括三个部分: 分词子系统、句法语义子系统、总控部分。在总控部分的协调下, 分词子系统可以获得有关词、句子等的句法和语义信息来对分词歧义进行判断, 即它模拟了人对句子的理解过程。这种分词方法需要使用大量的语言知识和信息。由于汉语语言知识的笼统、复杂性, 难以将各种语言信息组织成机器可直接读取的形式, 因此目前基于理解的分词系统还处在试验阶段。

2.1.3.3 常见中文分词技术缺陷分析

有了成熟的分词算法, 并不能很容易就解决中文分词的问题。因为中文是一种十分复杂的语言, 让计算机理解中文语言更是困难。在常见中文算法中, 有一些问题一直没有完全突破[23~25]。

1) 效率低

效率低是最大匹配法分词必然会来的问题。即使我们可以将字长设成相当短, 例如 5 (注意, 我们不能再缩短字长了, 毕竟字长为 5 以上的词太多了, 我们不能牺牲分词的准确), 然而当我们的最大词长为 2 时, 至少有 3 次的匹配算法是浪费掉的。回想一下算法书里提到的最简单的字符匹配与 KMP 算法之间天差地别的效率, 我们知道通过某种方法, 这些浪费的掉的匹配时间是可以补回来的。

2) 分词歧义

中文是很复杂的语言，它的表达方式如此之多，语法文法如此精妙，机械的电脑是很难理解这么复杂的语言，因此它必然会带来歧义性，以下是两个简单的例子：

A. “有意见分歧”（正向最大匹配和逆向最大匹配结果不同）

有意/ 见/分歧/

有/ 意见/分歧/

B. “结合成分子时”（正向最大匹配和逆向最大匹配结果相同）

结合/ 成分/ 子时/由于词的歧义性使我们在使用最大匹配法分词会产生错误的结果，而且使用正向分词与逆向分词往往会产生截然不同的结果。尽管使用回溯法或计算计算词的使用频率，可以使出现歧义的可能性减少，但是我们知道，这样的结果是不可避免的，因为中文的变化实在太多了。

3) 新词识别

新词，专业术语称为未登录词[18]。也就是那些在字典中都没有收录过，但又确实能称为词的那些词。最典型的是人名，人可以很容易理解句子“李四去广州了”中，“李四”是个词，因为是一个人的名字，但要是让计算机去识别就困难了。如果把“李四”作为一个词收录到字典中去，全世界有那么多名字，而且每时每刻都有新增的人名，收录这些人本身就是一项巨大的工程。

新词中除了人名以外，还有机构名、地名、产品名、商标名、简称、省略语等都是很难处理的问题，而且这些又正好是人们经常使用的词，因此对于搜索引擎来说，分词系统中的新词识别十分重要。目前新词识别准确率已经成为评价一个分词系统好坏的重要标志之一。

综上所述，如何解决中文分词的效率、歧义及新词问题是本文要解决的关键问题。本文结合以往的分词算法，提出一套新的分词处理方法。很好的解决了效率、歧义及新词问题。

2.2 P2P 搜索技术概述

2.2.1 P2P 搜索技术概述

第二代搜索引擎虽然比第一代在搜索速度，针对多语言信息的扩展方面有所改进，在以自然语言为查询语言的方面也做了一些探索，然而随着互联网的快速发展，网络中的数字化信息和人们所需获取信息能力之间的矛盾日益突出。一般的搜索引擎只能查看 HTML

格式, 因为搜索引擎的 Spiders 蜘蛛程序只能接受这种格式的网页, 这就代表着, 在局域网中, 其他格式的文件无法被外部的搜索引擎查询到。随着互联网技术的发展, P2P 技术应用到文件检索中, 通过共享硬盘上的文件, 目录乃至整个个硬盘, 用户搜索无需通过 Web 服务器, 不受信息文档格式的限制, 既可以达到传统搜索引擎无可比拟的深度[26]。

对比普通搜索引擎, P2P 搜索引擎具有先天性的优势[26]:

- 覆盖范围广

因为通过 P2P 技术, 节点共享制定目录, 甚至整个硬盘, 在理想状态下, P2P 搜索技术可以覆盖全部的互联网信息资源。

- 搜索速度快

因为 P2P 技术间的查询直接在节点间进行, 无需通过中央服务器, 不存在性能瓶颈, 相比传统搜索引擎查询速度更快。

2.2.2 P2P 模式与 C/S 模式的比较

传统的 C/S 模型网络如图 2-1 所示, 有一台或多台服务器以及大量的客户机。服务器配备大容量存储器并安装数据库系统, 用于数据的存放和数据检索; 客户端安装专用的软件, 负责数据的输入、运算和输出。客户机和服务器都是独立的计算机。客户端和服务器的角色在模型中是固定不变的。

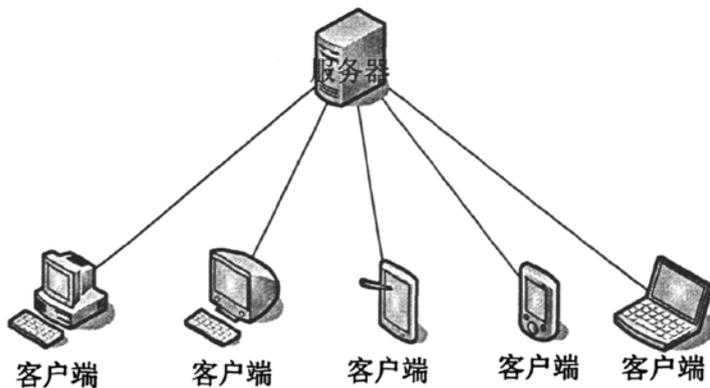


图 2-1 传统 C/S 模型网络图

由图 2-1 知, 可以看出 C/S 模型具有如下特点:

1. 信息集中发布, 集中存储, 集中管理。服务器控制所有信息, 而客户端只负责读取, 反馈。这就方便了数据的管理和更新。
2. 服务器的硬件性能及网络状况决定 C/S 模型能服务客户端的容量, 及网络的稳定

性能。

3. C/S 模型的中央集中特性决定了网络的稳定性差,安全性能差,一定服务器出现问题,网络马上崩溃,所有客户端将无法获取数据及进行其他操作。在一个网络中可能存在成万上亿的客户端同时操作同一数据,无论服务器端硬件设施及网络带宽状态无何,面对海量的访问,中心服务器始终存在性能瓶颈,而且这个问题是由其先天决定无法解决的。
4. 安全性能差,节点间的通讯,必须经过中心服务器,如果黑客破解中心服务器安全密匙,整个网络的信息都暴露在黑客眼前。

而 P2P 网络模型是非中心化的,每个节点既是服务器,又是客户端,客户端越多服务器越多,它与集中控制的 C/S 模型有明显的区别。简单的 P2P 模型网络如图 2-2 所示,具备如下几个方面的优点。

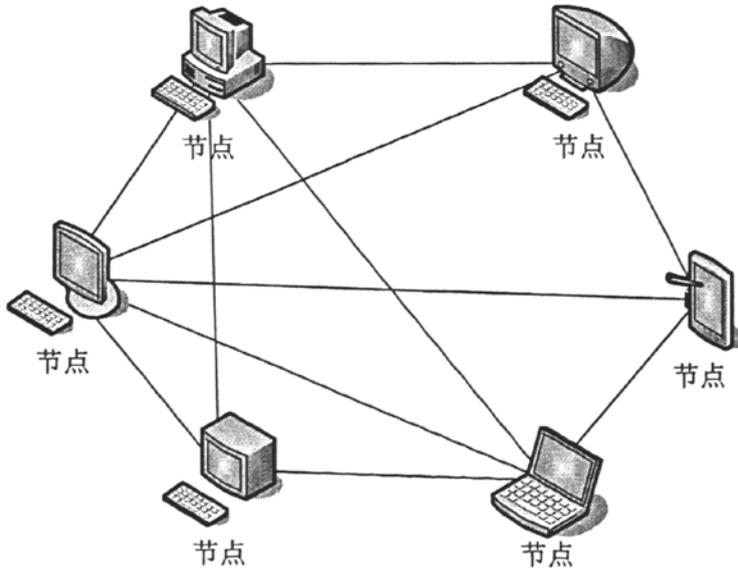


图 2-2 P2P 模型网络图

1. 信息分散存储,每个节点只负责保存自己信息或者部分邻居节点部分信息索引。整个网络中并不存在一个集中的中心服务器负责所有信息的存储控制和管理。
2. 网络中节点数目从一定程度决定网络性能的高低,如果有好的算法,节点越多,网络安全性,稳定性越好。这与 C/S 模型是相反的。就像前文描述的 BT 一样,节点越多,可以提供下载共享的文件片段来源越多,下载速度越快。
3. 节点在网络中的地位一方面是客户端,另外一方面是服务器。就是说节点一边享

受其他节点的服务，同时为其他节点提供服务。

4. P2P 网络可扩展性好，P2P 的网络结构决定了新节点很容易加入网络并成为其中正式的一员。从而进一步扩大增强网络的服务能力和性能。
5. 网络压力均衡，比较稳定。因为网络中的信息计算，存储通过较好的算法控制可以均衡的分配到网络中的每个节点上。充分利用了网络中空闲节点的 CPU, 及存储资源。某个节点的加入与退出并不会给网络带来多大的影响。
6. 由于网络中节点直接通讯，无需通过中心服务器信息中转，网络的安全系数大大提高。

通过对 C/S 模型及简单 P2P 模型的介绍分析，通过比较可以得出 P2P 模型的优缺点：

1. P2P 网络具有天生的资源高效利用的特点。
2. P2P 网络并不存在明显的性能和网络瓶颈，C/S 模型中一旦服务器节点崩溃，整个网络即将瘫痪。而在 P2P 网络中节点的地位是平等的，一个节点的失效并没有多大的影响。随着节点的增加，在 P2P 中网络的服务性能是慢慢增加的，而 C/S 模型刚好与此相反。
3. P2P 网络具有信息安全性高，容错能力强，传输速度快的特点。由于不经过中心服务器中转，某个节点被控制并不意味着整个网络的不安全。同时，节点间通过较好的路由算法直接或者间接的传输信息，具有快速传播信息的特点。

同时 P2P 模型对比 C/S 模型也有许多不足的地方。

1. P2P 模型缺乏统一集中的中心管理机制，信息不容易控制。
2. 由于节点间信息的随意交互，信息真伪及安全无法得到有效的保证。
3. 由于节点在网络中的随意加入退出，造成 P2P 网络某种程度上的不稳定。

节点间信息的交互，到目前为止还有出现绝对意义上高效的算法，造成了一定程度上的带宽重复，从而造成资源浪费。但是 P2P 搜索引擎处于起步阶段，各种技术不是完全成熟，下面我们进一步分析常见 P2P 搜索技术的网络结构。

2.2.3 常见 P2P 搜索技术网络结构分析

通过对文献的充分研究，当前 P2P 搜索技术所采用的网络模型主要有四种：中心化网络模型、全分布非结构化模型、半分布式模型和全分布结构化模型。本节将详细介绍这四种模型结构[26~52]。

2.2.3.1 中心化网络结构

中心化拓扑也叫集中目录式网络,其结构如图 2-3 所示,其典型代表就是 Napster[42],系统中所有节点直接连接到中央服务器,其搜索流程如下:

1. 节点加入网络,然后向中央服务器发布自己的节点存储的文件信息。中央服务器记录该节点的资源信息。
2. 节点发起搜索请求,请求将直接发送到中央服务器,中央服务器搜索资源数据库,定位到资源节点,返回结果给请求节点。
3. 请求节点与资源节点建立直接连接,文件传输在这两个节点间传输,不通过中央服务器。

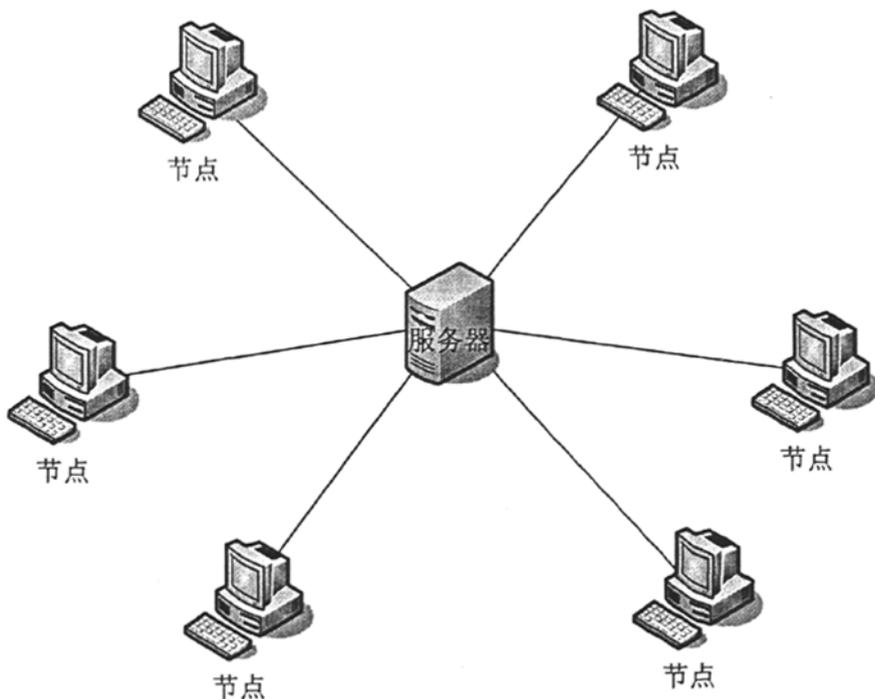


图 2-3 P2P 集中目录式网络结构

Napster 是最早出现的 P2P 系统之一,并在短期内迅速成长起来。Napster 实质上并非纯粹的 P2P 系统,它通过一个中央服务器保存所有 Napster 用户上传的音乐文件索引和存放位置的信息。当某个用户需要某个音乐文件时,首先连接到 Napster 服务器,在服务器进行检索,并由服务器返回存有该文件的用户信息;再由请求者直接连到文件的所有者传输文件。

在 Napster 模型中,一群高性能的中央服务器保存着网络中所有活动对等计算机共享

资源的目录信息。当需要查询某个文件时,对等机会向一台中央服务器发出文件查询请求。中央服务器进行相应的检索和查询后,会返回符合查询要求的对等机地址信息列表。查询发起对等机接收到应答后,会根据网络流量和延迟等信息进行选择,和合适的对等机建立连接,并开始文件传输。

这种对等网络模型存在很多问题,主要表现为:

1. 中央服务器的瘫痪容易导致整个网络的崩溃,可靠性和安全性较低。
2. 随着网络规模的扩大,对中央索引服务器进行维护和更新的费用将急剧增加,所需成本过高。
3. 中央服务器的存在引起共享资源在版权问题上的纠纷,并因此被攻击为非纯粹意义上的 P2P 网络模型。

对小型网络而言,集中目录式模型在管理和控制方面占一定优势。但鉴于其存在的种种缺陷,该模型并不适合大型网络应用。

2.2.3.2 全分布非结构化网络结构

全分布非结构化拓扑也叫纯 P2P 网络结构,其模型如图 2-4 所示,其典型代表为 Gnutella[43, 44]。该类型网络中不存在中央服务器。每个节点地位平等,各自存储自己节点资源索引数据。每个节点既可以向网络中其他节点请求资源,同时又为其他节点提供请求服务。文件的搜索和传输在各个节点间通过不同的算法通讯。

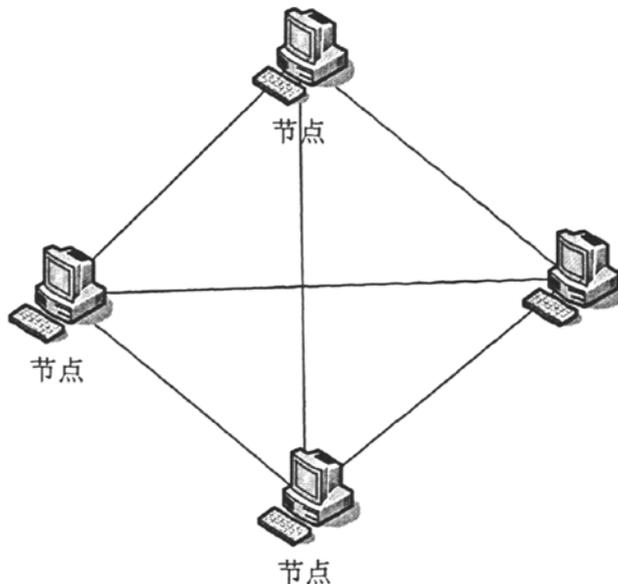


图 2-4 简单纯 P2P 模式网络结构

按照搜索策略,可以分为两大类:盲目搜索和启发式搜索。盲目搜索通过在网络中传播查询信息并且把这些信息不断扩散给每个节点。通过这种洪泛方式来搜索想要的资源。而启发式搜索在搜索的过程中利用一些已有的信息来辅助查找过程。由于信息搜索对资源的存储有一些知识,所以信息搜索能够比较快的找到资源。

Gnutella 是一个 P2P 文件共享系统,它和 Napster 最大的区别在于 Gnutella 是纯粹的 P2P 系统,没有索引服务器,它采用了基于完全随机图的洪泛(Flooding)发现和随机转发(Random Walker)机制。为了控制搜索消息的传输,通过 TTL (Time To Live)的减值来实现。

在 Gnutella 分布式对等网络模型 N 中,每一个联网计算机在功能上都是对等的,既是客户机同时又是服务器,所以被称为对等机(Servent, Server+Client 的组合)。

随着联网节点的不断增多,网络规模不断扩大,通过这种洪泛方式定位对等点的方法将造成网络流量急剧增加,从而导致网络中部分低带宽节点因网络资源过载而失效。所以在初期的 Gnutella 网络中,存在比较严重的分区,断链现象。也就是说,一个查询访问只能在网络的很小一部分进行,因此网络的可扩展性不好。所以,解决 Gnutella 网络的可扩展性对该网络的进一步发展至关重要。

由于没有确定拓扑结构的支持,非结构化网络无法保证资源发现的效率。即使需要查找的目的结点存在发现也有可能失败。由于采用 TTL (Time-to-Live)、洪泛(Flooding)、随机漫步或有选择转发算法,因此直径不可控,可扩展性较差。

因此发现的准确性和可扩展性是纯 P2P 网络结构面临的两个重要问题。

2.2.3.3 半分布网络结构

半分布式拓扑也叫混合式 P2P 网络结构,其网络结构如图 2-5 所示,它吸取了中心化结构和全分布式非结构化拓扑的优点,选择性能较高(处理、存储、带宽等方面性能)的结点作为超级点(英文文献中多称作: SuperNodes, Hubs),在各个超级点上存储了系统中其他部分结点的信息,发现算法仅在超级点之间转发,超级点再将查询请求转发给适当的叶子结点。半分布式结构也是一个层次式结构,超级点之间构成一个高速转发层,超级点和所负责的普通结点构成若干层次。最典型的案例就是 KaZaa[45]。

KaZaa 是现在全世界流行的几款 p2p 软件之一。根据 CA 公司统计,全球 KaZaa 的下载量超过 2.5 亿次。使用 KaZaa 软件进行文件传输消耗了互联网 40%的带宽。之所以它如此

的成功,是因为它结合了 Napster 和 Gnutella 共同的优点。从结构上来说,它使用了 Gnutella 的全分布式的结构,这样可以使系统更好的扩展,因为它无需中央索引服务器存储文件名,它是自动的把性能好的机器成为 SuperNode,它存储着离它最近的叶子节点的文件信息,这些 SuperNode,再连通起来形成一个 Overlay Network[38]。由于 SuperNode 的索引功能,使搜索效率大大提高。

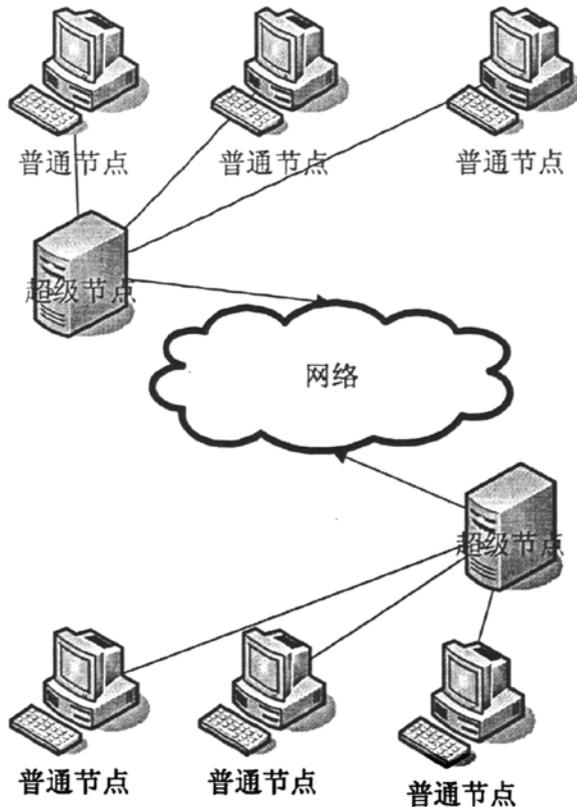


图 2-5 混合式网络模型

半分布式结构的优点是性能、可扩展性较好,较容易管理。但它依然存在问题:对超级节点依赖性大,一旦超级节点崩溃或者失效,局部网络将变的无效;同时因为由于超级节点的性能,很容易受到攻击;其网络模型容错性也比较差。

2.2.3.4 全分布结构化网络结构

结构化 P2P 网络结构系统通过 DHT(DHT, Distributed Hash Table)[46~54]技术实现搜索。DHT 的原理就是把系统中每个文件索引通过特定的规则用 $\langle K, V \rangle$ 来标识,其中 K 称为关键字, V 是存储该资源文件的节点信息。系统中所有文件索引条目组成一张很大的文件索引哈希表。某个节点请求资源时,只要对该哈希表进行查找就可以找出存储该资源的节

点实际地址。由于整个系统的哈希表过于庞大，因此将该哈希表按照特定规则分成小块，每个节点负责存储维护其中一块，某节点发送搜索请求的时候只要把搜索请求发送到相应的节点就可以找到相应的资源。下面我们对几种使用 DHT 技术的主要网络技术进行讨论。

Tapestry[50]提供了一个分布式容错查找和路由基础平台，在此平台基础之上，可以开发各种 P2P 应用(OceanStore[51]即是此平台上的一个应用)。Tapestry 的思想来源于 Plaxton。在 Plaxton 中，结点使用自己所知道的邻近结点表，按照目的 ID 来逐步传递消息。Tapestry 基于 Plaxton 的思想，加入了容错机制，从而可适应 P2P 的动态变化的特点。OceanStore 是以 Tapestry 为路由和查找基础设施的 P2P 平台。它是一个适合于全球数据存储的 P2P 应用系统。任何用户均可以加入 OceanStore 系统，或者共享自己的存储空间，或者使用该系统中的资源。通过使用复制和缓存技术，OceanStore 可提高查找的效率。最近，Tapstry 为适应 P2P 网络的动态特性，作了很多改进，增加了额外的机制实现了网络的软状态 (soft state)，并提供了自组织、鲁棒性、可扩展性和动态适应性，当网络高负载且有失效结点时候性能有限降低，消除了对全局信息的依赖、根结点易失效和弹性 (resilience) 差的问题。

Pastry[52]是微软研究院提出的可扩展的分布式对象定位和路由协议，可用于构建大规模的 P2P 系统。在 Pastry 中，每个结点分配一个 128 位的结点标识符号 (nodeID)，所有的结点标识符形成了一个环形的 nodeID 空间，范围从 0 到 $2^{128} - 1$ ，结点加入系统时通过散列结点 IP 地址在 128 位 nodeID 空间中随机分配。

Chord[54]是 MIT 的一个基于 P2P 的项目，Chord 项目的目标是提供一个适合于 P2P 环境的分布式资源发现服务，它通过使用 DHT 技术使得发现指定对象只需要维护 $O(\log N)$ 长度的路由表。

在 Chord 技术中，网络结点按照一定的方式分配一个唯一结点标识符 (Node ID)，资源对象通过散列运算产生一个唯一的资源标识符 (Object ID)，且该资源将存储在结点 ID 与之相等或者相近的结点上。需要查找该资源时，采用同样的方法可定位到存储该资源的结点。因此，Chord 的主要贡献是提出了一个分布式查找协议，该协议可将指定的关键字 (Key) 映射到对应的结点 (Node)。从算法来看，Chord 是相容散列算法的变体。MIT 的 GRID 和 RON 项目则提出了在分布式广域网中实施查找资源的系统框架。AT&T ACIRI 中心的 CAN (Content Addressable Networks) [55] 项目独特之处在于采用多维的标识符空间来实现分布式散列算法。CAN 将所有结点映射到一个 n 维的笛卡尔空间中，并为每个结点尽可能均匀的分配一块区域。CAN 采用的散列函数通过对 (key, value) 对中的 key 进行散列运算，得到笛卡尔空间中的一个点，并将 (key, value) 对存储在拥有该点所在区域的结点

内。CAN 采用的路由算法相当直接和简单，知道目标点的坐标后，就将请求传给当前结点四邻中坐标最接近目标点的结点。CAN 是一个具有良好可扩展性的系统，给定 N 个结点，系统维数为 d ，则路由路径长度为 $O(n^{1/d})$ ，每结点维护的路由表信息和网络规模无关为 $O(d)$ 。

综合分析知道结构化 P2P 网络结构存在两个严重的问题[46-54]:

1. 是 DHT 的维护机制较为复杂，尤其是结点频繁加入退出造成的网络波动(Churn)会极大增加 DHT 的维护代价。
2. 其次是 DHT 仅支持精确关键词匹配查询，无法支持内容/语义等复杂查询。

2.2.3.5 常见 P2P 搜索技术网络结构综合性能分析

通过对常见 P2P 搜索技术及其网络结构的分析，总结出各种网络结构模型在可扩展性、可靠性、可维护性、发现算法效率、复杂查询支持等标准的综合性能如表 2-1 所示，从表 2-1 可以看出全分布式结构化拓扑网络无法支持复杂搜索，而中心化拓扑的可扩展性可靠性最差，一旦节点数量达到一定规模，系统将无法保持稳定可靠。全分布非结构化拓扑可扩展性很差，同样不能适应大规模的网络节点。半分布式拓扑各项性能处于中等。因此，如何实现一个可扩展、可靠、可维护、发现算法效率高及支持复杂搜索的网络拓扑结构也是本章研究的一个核心内容。

表 2-1 各种网络拓扑结构综合性能比较

比较标准	中心化拓扑	全分布非结构化拓扑	全分布结构化拓扑	半分布拓扑
可扩展性	差	差	好	中
可靠性	差	好	好	中
可维护性	最好	最好	好	中
发现算法效率	最高	中	高	中
复杂查询	支持	支持	不支持	支持

2.3 本章小结

本章开篇以中文搜索引擎技术为先，介绍了中文搜索引擎的发展历程，对中文搜索引擎的基本技术进行了介绍。着重对影响中文搜索引擎搜索结果的关键技术——中文分词做了重点分析，对中文分词的概念，目前常见的中文分词技术模型及其存在的缺陷进行了总结。接着按照 P2P 网络拓扑结构对目前常见的 P2P 搜索软件技术进行了分类介绍及分析重

点指出了各种网络模型存在的不足，最后总结了常见 P2P 搜索技术所采用的网络拓扑结构的优缺点。

第三章 P2PCSE 系统总体设计

第二章相关技术的分析为本章出的 P2PCSE 系统总体设计提供了理论基础，同时也为 P2PCSE 系统中的关键技术提供了对比原型。综合中文搜索引擎及 P2P 搜索技术，本章首先提出 P2PCSE 系统总体设计，根据总体设计，制定了具体的设计要求，对 P2PCSE 系统进行了具体的功能模块划分，在此基础上详细描述了系统流程结构，对主要通讯协议进行了详细设计，最后对系统关键技术：基于 X-树结构中文分词算法的全文检索及 P2P 三层混合式半分布网络结构进行了定义和设计。

3.1 系统总体设计

P2PCSE 系统应兼备 P2P 系统良好的可扩展性及中文搜索引擎技各方面的特点，鉴于项目目前期的对各方面情况的综合分析，本节提出了 P2PCSE 系统总体设计如图 3-1 所示。

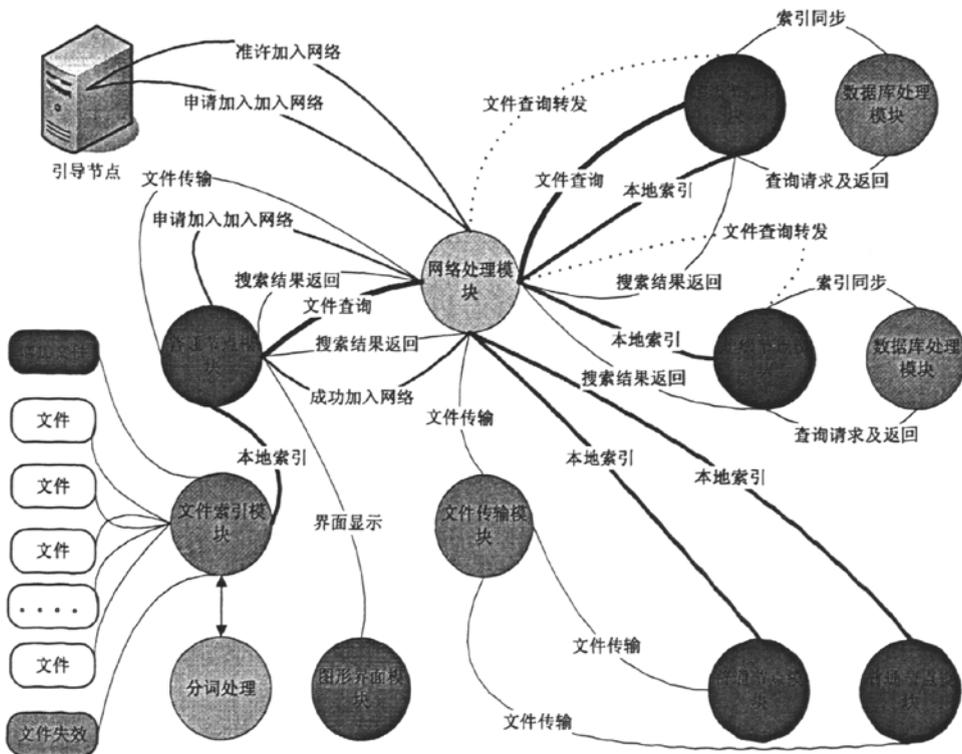


图 3-1 P2PCSE 系统总体设计图

P2PCSE 总体设计基于两大关键部分：全文检索及网络结构。本文第四章、第五章将分别对其一一详细分析及实现。其中全文检索实现了系统完整索引 P2PCSE 网络所有共享数

据，并在普通节点新增文件及文件失效的情况下保持索引数据与网络真实数据同步更新。P2PCSE 系统基于一种改进的半分布式混合三层网络结构，该网络结构模型保证了系统在大规模的并发请求下依然能保持网络稳定可靠，同时避免传统半分布式网络结构在超级节点崩溃情况下导致网络局部失效。

按照功能可将节点划分为三种类型：引导节点、超级节点及普通节点。普通节点向引导节点发送申请加入网络的请求，在成功加入网络后索引本地共享文件，与 P2PCSE 系统分配的超级节点同步索引数据。超级节点与数据库处理模块为普通节点提供共享文件索引数据的同步、查询功能。普通节点的文件处理、索引、同步及超级节点的索引数据储存、查询服务共同构成了 P2PCSE 系统的全文检索系统。当 P2PCSE 系统中普通节点搜索网络资源时，在基于系统网络结构的基础上通过特定的通讯协议向超级节点发送搜索请求，超级节点检索本地数据库并与网络中其他超级节点交互，并将搜索结果返回给请求节点。请求节点在接受到搜索结果后与结果数据所在普通节点建立连接。获取相应数据。

通过对第二章相关技术的分析，本文总结出 P2PCSE 系统总体设计目标如下：

1. 具备优异的 P2P 网络结构, 系统稳定健壮易于扩展, 新节点能快速加入网络, 网络不因为个别节点的失效而影响整个系统的稳定及搜索效率。
2. 具备先进全文检索技术, 使得 P2PCSE 系统可以准确快速全面的索引数据, 提供查询服务。
3. 搜索应尽可能准确理解请求, 尽可能多按搜索请求返回符合请求的结果数据。
4. 搜索应快速高效, 在大规模的并发搜索请求情况下不对网络造成很大的压力。

3.2 系统功能模块划分

P2PCSE 系统按照功能划为为分词处理模块、文件索引模块、普通节点模块、超级节点模块、数据库处理模块、文件传输模块、网络处理模块、图形界面模块等八大模块。

其中各模块具体功能如表 3-2 所示。

表 3-2 P2PCSE 功能模块表

功能模块名称	负责功能
分词处理模块	负责处理 P2PCSE 系统中共享文件, 将原始数据分解成中文, 英文, 中英文混合式分别处理。其中尤其以处理中文数据为重点。中文数据将分解为以单独的句子, 对句子利用中文分词进行解析、索引供文件索引模块调用。

文件索引模块	按照文件类型表对文件进行分类索引, 利用分词模块重点处理中文数据。通过网络处理模块将数据发送给自治簇超级节点保存。
普通节点模块	调用分词处理, 文件索引, 文件传输, 图形界面, 网络等模块组织成为一个具备完整功能的普通节点。
超级节点模块	通过判断节点硬件及网络性能, 将符合条件的节点通告引导节点供其分配, 成为某自治簇的超级节点。超级节点不提供文件共享服务。只存储自治簇普通节点文件索引数据, 并保存到数据库处理模块。对节点请求和转发请求提供查询服务, 同时对某些局域网节点提供文件中转服务。
数据库处理模块	保存普通节点的经过分词模块处理后的文件索引数据。配合超级节点模块对其他节点请求提供增加、删除、更新索引数据及查询服务。
文件传输模块	主要实现文件传输功能。
网络处理模块	基于 UDP 封装各种 P2PCSE 自定义通讯协议, 为上述模块提供网络通讯服务。
图形界面模块	提供一个友好、人性化的界面给用户, 提供初始化, 共享设置、文件索引进度、节点列表、文件搜索结果等并提供交互服务。

P2PCSE 系统节点功能模块如图 3-4 所示:

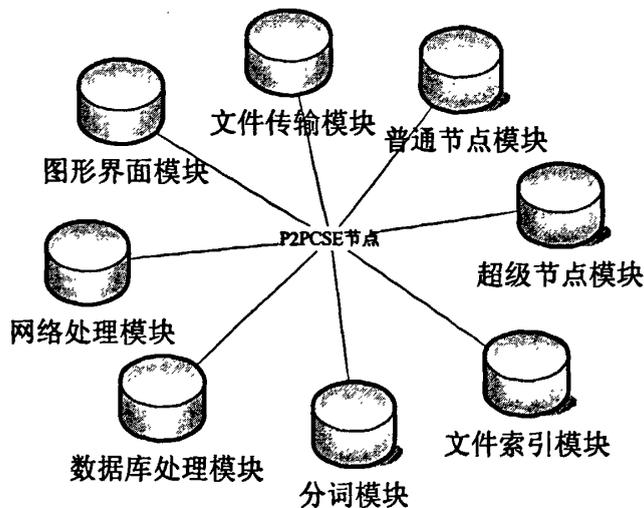


图 3-4 P2PCSE 系统节点模块图

3.3 系统流程结构

在 P2PCSE 系统中, 某结点从初始化到发送搜索请求直到超级节点返回搜索结果。其一般流程如图 3-5 所示。

- 1) 节点首先运行 P2PCSE 系统, 如果程序检测到系统初次运行, 则要求设置共享目录。
- 2) 程序调用 P2PCSE 索引模块。对其中的中文文件调用中文分词算法进行关键字解析索引。
- 3) 在 2) 的同时向系统引导节点发送加入网络的请求, 得到允许后节点成功加入网络。然后将共享文件索引数据与该节点所在自治簇超级节点同步。
- 4) 当用户发起某关键字搜索请求后, 节点首先将请求发送给该自治簇超级节点。在搜索不充分的情况下, 超级节点通过限定转发数目将请求随机转发给网络中其他超级节点。搜索完成后超级节点将数据返回给请求源节点。

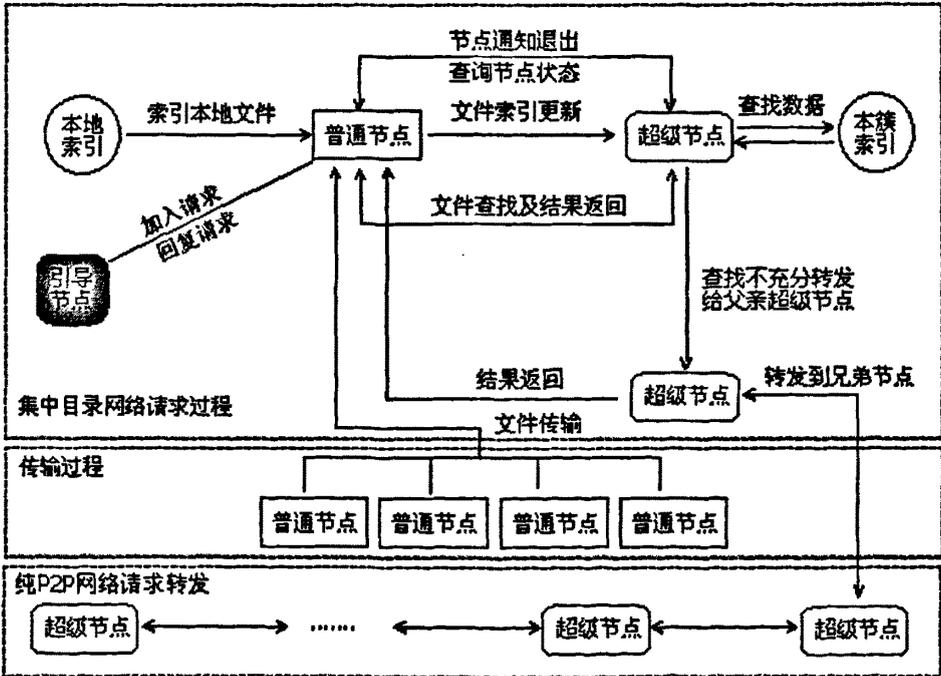
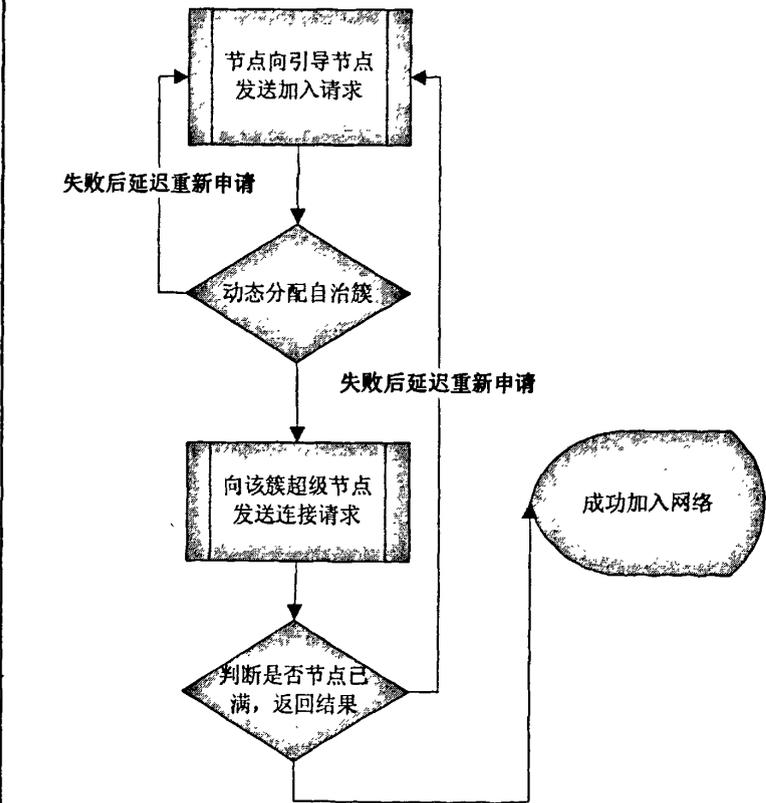


图 3-5 P2PCSE 系统流程图

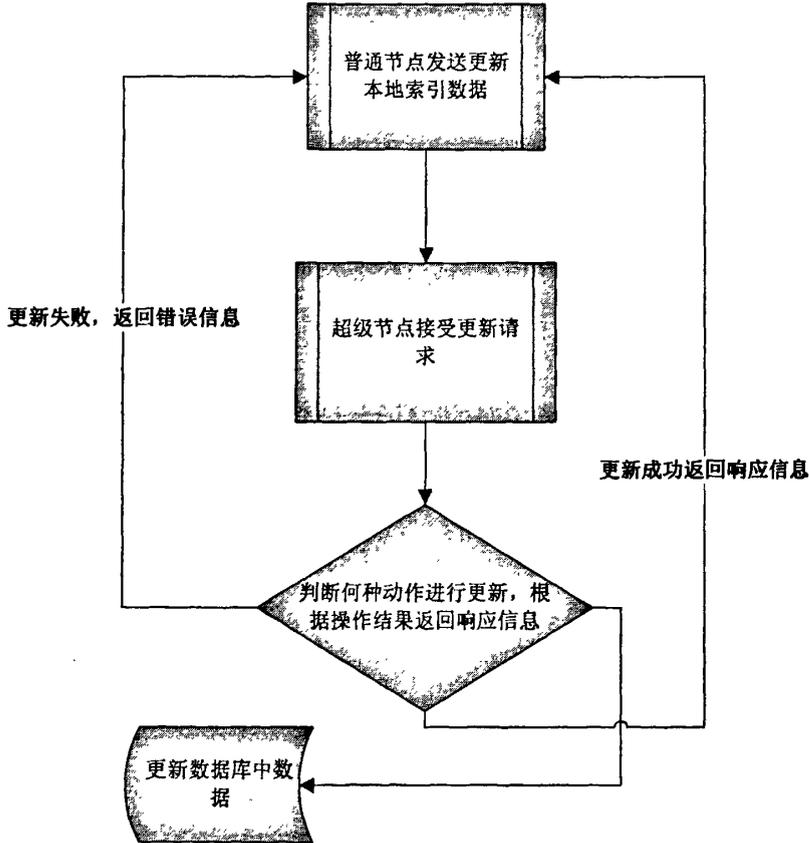
3.4 系统主要通讯协议设计

P2PCSE 系统协议采用基于 UDP 的通讯模式, 节点通过建立多线程 SOCKET 服务机制, 用于发送接受数据, 主要协议如表 3-3 所示。

表 3-3 P2PCSE 系统主要协议

协议名称	REPLY
协议参数	Data: 不同情况下返回不同结果;
协议描述	响应节点请求, 返回请求后的处理结果。
协议名称	ERROR
协议参数	Data: 不同情况下返回不同结果;
协议描述	处理节点请求, 参数表示动作未按预定期望执行后的错误信息。
协议名称	PEERJOIN
协议参数	PEERID: 节点网络 ID;
协议描述	节点请求加入网络向引导节点发送加入请求。引导节点根据节点参数将节点指定给某自治簇超级节点。请求节点接受返回数据后向对应超级节点发送请求加入命令, 建立连接。
协议流程	 <pre> graph TD A[节点向引导节点发送加入请求] --> B{动态分配自治簇} B -- "失败后延迟重新申请" --> A B --> C[向该簇超级节点发送连接请求] C --> D{判断是否节点已满, 返回结果} D -- "失败后延迟重新申请" --> A D --> E([成功加入网络]) </pre>

协议名称	PEERQUIT
协议参数	PEERID: 退出节点网络 ID;
协议描述	超级节点接受到此请求后, 将该节点标志为失效节点。
协议名称	PING
协议参数	NULL;
协议描述	用于判断某节点是否为有效节点, 节点在接受该请求后立即给请求节点返回响应。否则超级节点在超时后设置节点为失效。
协议流程	<pre> graph TD A[发送PING命令] --> B[接受PING命令] B --> C{立即返回} C --> A B -- "超时无响应节点无效" --> A B -- "有效节点立即返回响应" --> C </pre>
协议名称	SNMSG
协议参数	PEERID: 发送节点网络 ID; TPEERID: 目标节点网络 ID; MSG: 信息内容;
协议描述	普通节点间文字信息通信。某节点发送文字信息及自身 PEERID 给目标节点, 目标节点在接受到该信息后, 显示信息内容及来源。

协议名称	SYNCFILEDATA
协议参数	PEERID: 普通节点网络 ID; ACTION: [ADD UPDATE DELETE]动作类型, 分别对应: 增加、更新、失效三种动作; FILE_DESCRIPTION: 文件索引具体数据;
协议描述	处理普通节点索引数据同步请求, 普通节点向超级节点更新节点包含共享文件索引数据。
协议流程	 <pre> graph TD A[普通节点发送更新本地索引数据] --> B[超级节点接受更新请求] B --> C{判断何种动作进行更新, 根据操作结果返回响应信息} C -- "更新成功返回响应信息" --> D[更新数据库中的数据] C -- "更新失败, 返回错误信息" --> A </pre>
协议名称	QUERY
协议参数	PEERID: 请求节点网络 ID; KEY: 经过分词算法处理的分词单位; MINRECSET: 最少记录数; STIMEOUT: 超时限制;

协议描述	超级节点尝试查询并返回查询结果，在查询不充分的情况下根据 MINRECSET 限制多线程方式将请求随机转发给其他超级节点，其他超级节点在查询完成后返回数据给请求节点。
协议名称	QRESPONSE
协议参数	PEERID: 请求节点网络 ID; SNODEINFO: 返回搜索请求结果的超级节点信息; HPEERID: 目标文件所在节点网络 ID; SRESULT: 超级节点返回的搜索结果;
协议描述	普通节点接受超级节点搜索请求结果响应。

协议 QUERY 和 QRESPONSE 主要构成普通节点数据查询及超级节点结果返回，其流程如图 3-5 所示：

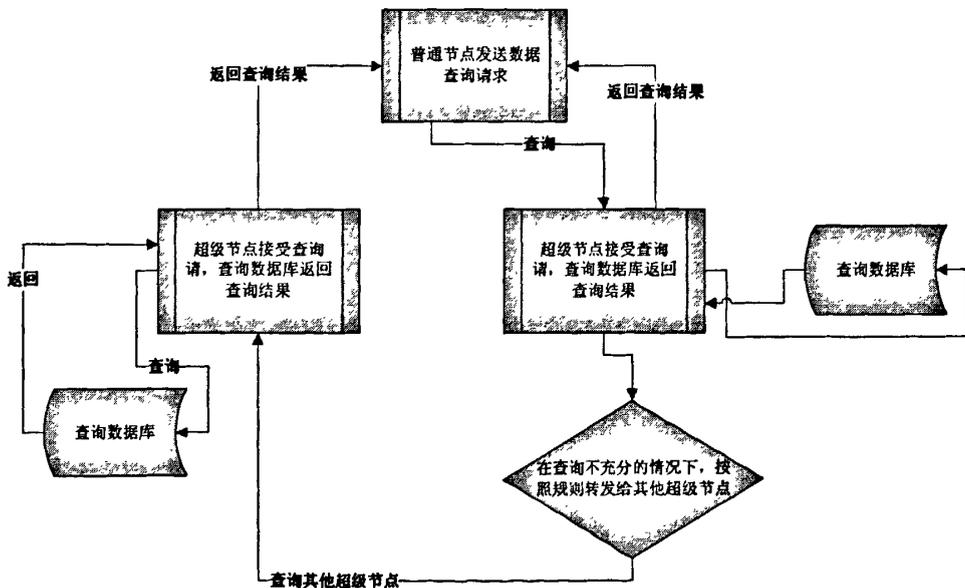


图 3-6 QUERY 及 QRESPONSE 复合流程图

3.5 系统关键技术分析

通过对 P2PCSE 总体系统设计目标的详细分析，本节对 P2PCSE 系统关键部分：全文检索和网络结构进行详细分析。并将在本文第四章、第五章进行详细实现。

3.5.1 全文检索分析

P2PCSE 系统为全文检索搜索引擎，其对应的文件索引为全文索引。其全文检索系统应具备先进的全文检索技术，可提供支持大规模的并发查询服务。通过第二章相关技术的分析，全文索引的关键技术是分词技术。要达到 P2PCSE 系统方案提出的目标，其关键是具备先进、高效的分词处理技术。

P2PCSE 系统分词处理模块按语言分为中文分词处理模块，英文分词处理模块及混合型分词处理模块。因为 P2PCSE 系统侧重对中文共享文件的搜索，P2PCSE 系统侧重中文分词的处理，对于英文数据及混合型数据只进行简单的分割匹配。

通过第二章相关技术研究中对中国搜索引擎中文分词技术的分析，要想 P2PCSE 系统达到系统方案预定的效果，P2PCSE 系统全文检索模块的中文分词处理应达到如下目标要求：

1) 高效

中文分词算法必须要高效，效率对于搜索引擎的重要性是至关重要的。而且搜索引擎面对的是海量的数据，效率的差别的影响可能会使结果查找的时间相差几个小时甚至几天。因此 P2PCSE 系统中的算法一定要比最大匹配法快，减少最大匹配法很多无效匹配过程，提高效率。

2) 无长度限制

最大匹配法的长度限制是很难解决的事情，从算法上分析基本没有可能找到词长与效率之间最佳的黄金分隔点。因此必须设计一种新的模式来避免有限制的匹配。

3) 歧义包容

一个好的搜索算法必须要或者最大可能的解决歧义包容问题，否则搜索的结果将是难以接受的。

4) 准确度高

搜索必须尽可能的理解请求者的本意，去掉那些无关的含义，才能算是一个好的搜索引擎。这也是目前所有搜索引擎急需处理的最重要问题。

5) 新词解析

随着社会的快速发展，新名词出现的速度越来越快，如何正确解析这些新名词，也是高效搜索引擎需要解决的一个问题。

3.5.2 网络结构分析

因为中文搜索引擎系统的特殊性，每个节点共享大量的数据，并采用了中文分词，文件分类等技术，精确定位数据，系统中任意节点对数据的查找及索引非常复杂，通过表 2-1 对 P2P 搜索技术采用的各种网络拓扑结构综合性能比较知道，采用中心化拓扑、全分布非结构化拓扑及全分布结构化拓扑网络结构是不现实的。

其中半分布拓扑网络结构具备了可扩展性、可维护性的特性，其可靠性、发现算法效率处于中等，支持复杂查询等特点。但是其各方面的效果都处于不理想的状态。并且某自治簇超级节点崩溃，该自治簇内所有节点处于孤立状态。

基于 P2PCSE 系统方案目标，P2PCSE 网络结构应达到如下具体要求：

- 1) 系统网络容易扩展、维护代价低、可靠稳定。
- 2) 系统网络资源发现效率高、支持复杂查询。

P2PCSE 系统网络将使用一种改进的半分布式网络结构，网络节点按照功能划分为普通节点、超级节点、引导节点。

3.6 本章小结

本章开篇在基于相关技术研究的基础上提出 P2PCSE 系统总体设计，接着对 P2PCSE 系统进行了具体模块划分，对各模块的功能做了定义，并从一般情况的角度描述了 P2PCSE 系统的流程，详细设计了 P2PCSE 系统的所用的主要通讯协议。对 P2PCSE 系统至关重要的两个部分——全文检索和网络结构进行了定义和分析，接下的第四章、第五章将详细对全文检索和网络结构进行设计和实现。

第四章 P2PCSE 系统全文检索实现

第三章对 P2PCSE 系统的关键部分——全文检索做了分析，为本章重点分析并实现系统全文检索子系统做了铺垫。本章首先介绍系统全文检索模型，按照功能将全文检索子系统划分为分词处理模块、文件处理模块、文件索引储存查询模块，并对其分别进行详细设计和实现。

4.1 系统全文检索模型

P2PCSE 全文检索系统模型如图 4-1 所示，负责 P2PCSE 系统网络中所有普通节点共享文件的数据索引储存、同步及为普通节点提供搜索服务。按照功能类别可将全文检索子系统划为三大主要模块：分词处理模块、文件处理模块、文件索引储存查询模块。

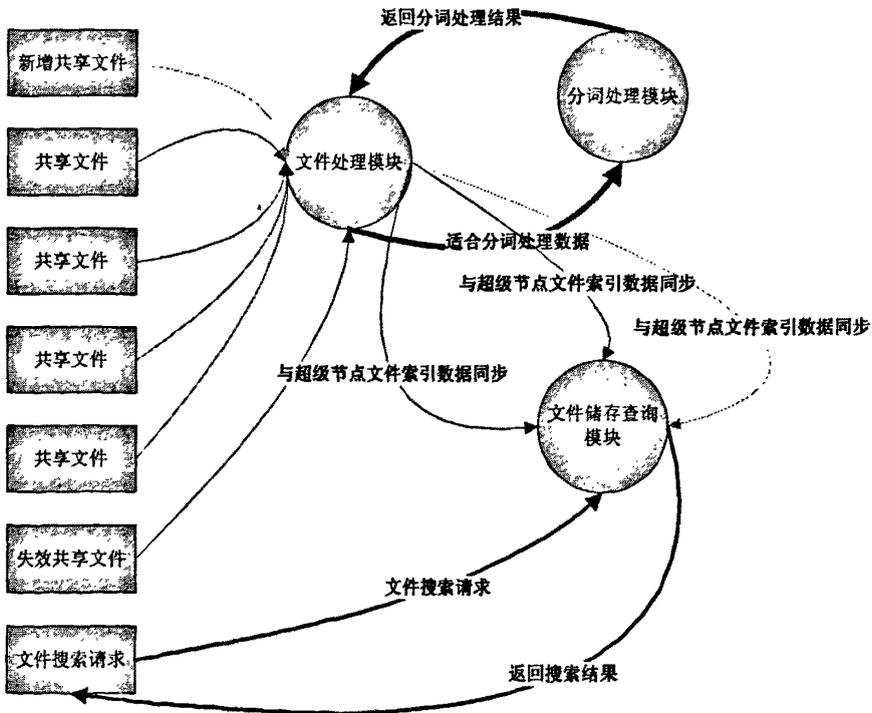


图 4-1 P2PCSE 系统全文检索模型

其中分词处理模块是全文检索子系统的核心，主要为普通节点索引本地共享文件提供索引工具。本节着重介绍了基于 X-树结构的中文分词处理算法。

文件处理模块主要在普通节点被调用，用于对普通节点共享文件夹下所有文件进行归类，根据文件归类表按规定调用相应文件类型处理方法读取文件，对特定数据调用分词处理模块进行进一步解析。

文件索引储存查询模块位于超级节点，保存普通节点同步到对应超级节点的所有索引数据，同时为 P2PCSE 系统中所有普通节点提供数据搜索服务。

本章将对三大模块进行具体定义和实现。

4.2 系统分词处理模块

P2PCSE 系统分词处理模块按语言分为中文分词处理 (CSECHSplitor)，英文分词处理 (CSEENSplitor) 及混合型分词处理 (CSEMIXSplitor)。P2PCSE 系统侧重中文数据处理，对英文及混合型数据只做简单分割，根据待索引数据长度按比例取出现次数最为频繁的分词单位。

根据第二章相关技术对中文分词处理技术的研究，本文提出了一种基于 X-树的中文分词算法，本节将从该中文分词模型词库数据结构、储存方式及查找算法三个方面重点介绍。

4.2.1 中文分词词库数据结构

P2PCSE 中文分词词库包含两个部分：普通词库、新词词库，在定义普通词库和新词词库前本节首先介绍一个自定义概念：X-树。

X-树：X-树（如图 4-1，类似 B-树，由此得名）定义如下：

一棵 X-树或者为空树，或者为满足如下条件的树：

- 1) 根节点指向初始结点（度为 1 的节点）地址；
- 2) 初始结点由一张附带指针的哈希表 [FHashTable] 构成，指针指向孩子结点开始地址，指针为空表示孩子结点为空。哈希表的每项元素由两个单元 HashKey 和 Fp 组成，HashKey 保存常见汉语分词单位首字的哈希值。Fp 保存孩子节点的开始地址；
- 3) 除初始结点和根结点外，其他所有结点包含如下信息：

$(n, R1, P1, F1, R2, P2, F2 \cdots Rn, Pn, Fn)$

其中 n 为 Ri 个数， $Ri (i=0, 1, 2 \cdots n)$ 为汉字编码且 $Ri < Ri+1 (i=0, 1, 2 \cdots n)$ ； $Pi (i=0, 1, 2 \cdots n)$ 为指向孩子结点的指针， Pi 为空表示孩子结点为空， $Fi (i=0, 1, 2 \cdots n)$ 为标志位，取值为 1 或者 0， $Fi=1$ 时，代表从初始节点到 Ri 可以构成有意义的分词单位。为 0 表示上述“分词单位”无意义。

4) 据统计除根结点和初始结点外, 每个结点所包含的 R 不超过一个常数 T(T<500);

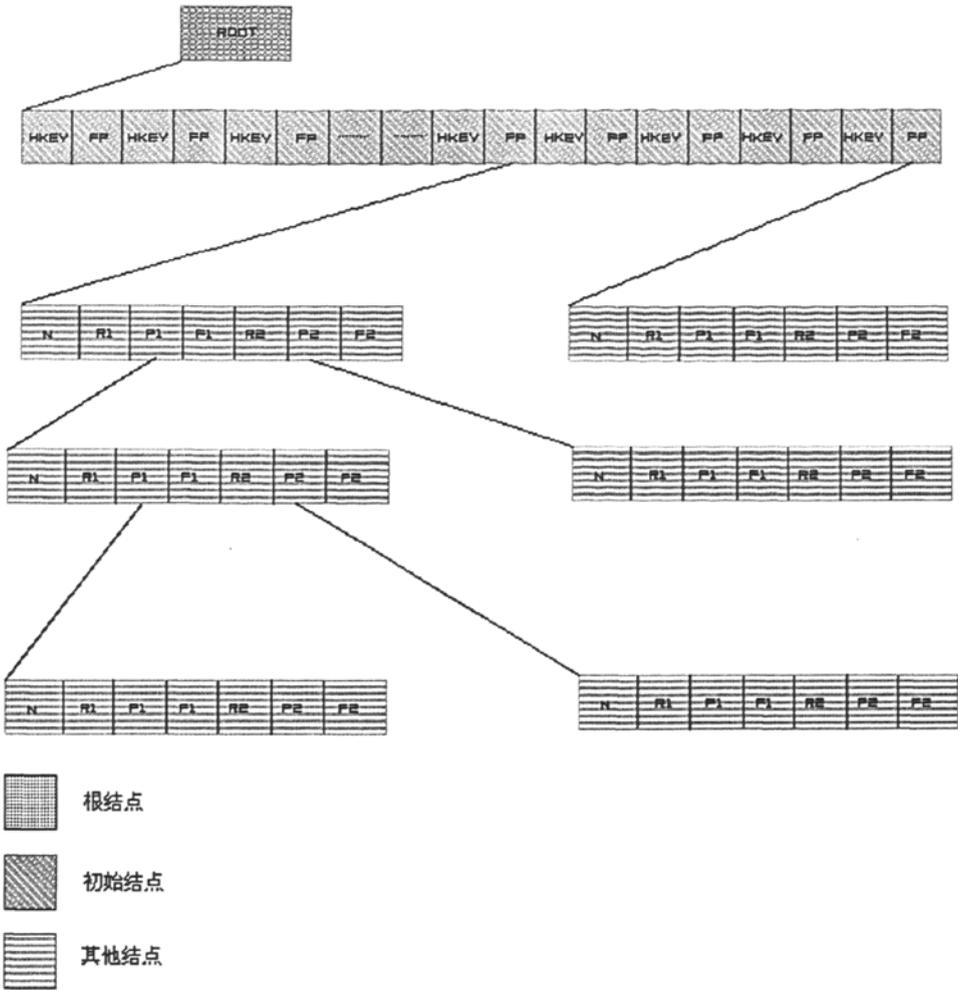


图 4-1 X-树结构图

把编码还原成汉字, 可以更方便的理解 X-树的结构, 如图 4-2 所示:

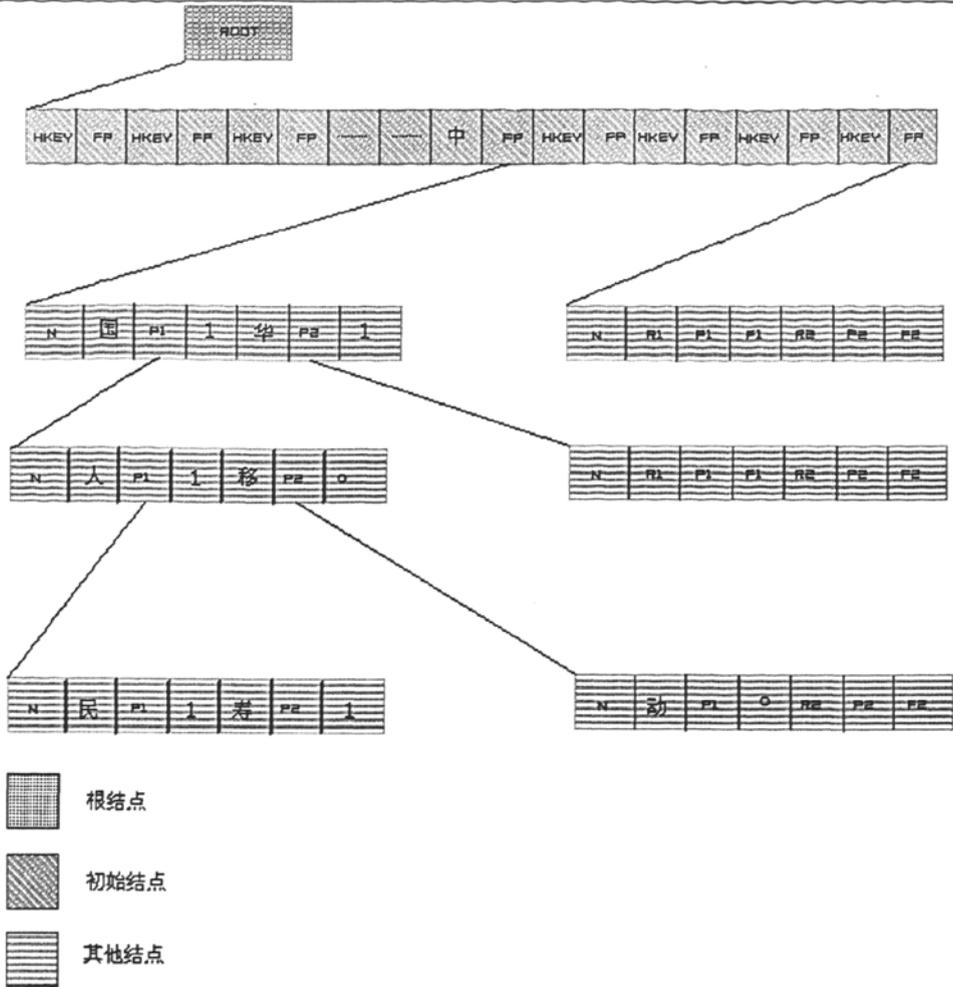


图 4-2 替代码对应汉字的 X-树结构图

有了 X-树的基础，下面给出普通词库和新词词库的定义及数据结构

普通词库：由所有有意义的中文分词单位组成的集合。在 P2PCSE 系统中，普通词库定义为基于 X-树的树型结构。

新词词库：新词词库记录近期出现的专有名词，比如“入世”，“下岗”，“房市”，“黄金周”，“买壳上市”，“五十年不变”，2009 年春晚大热的“小沈阳”等。新词词库采用 P2PCSE 新词词库哈希算法直接生成一张新词哈希表(NewWordsHashTable)。

4.2.2 中文分词词库储存及创建

中文分词普通词库的储存，采用关系数据库方法是很不理想。尤其是，关系数据库不理解对象。如果采用二进制模式存储，二进制到 X-树模型之间的转化是很大的开销，在 Python 语言中可以使用序列化过程 pickle 存储 Python 对象。可以将对象 pickle 成字符

串、磁盘上的文件或者任何类似于文件的对象，也可以将这些字符串、文件或任何类似于文件的对象 unpickle 成原来的对象。同时 pickle 是独立于操作系统的，在 Python 中 pickle 过程方法是向后兼容的。

通过 pickle 根据上文中的 X-树模型及替代表根据《汉语大词典》(约 38 万词)建立起系统中文分词普通词库，最后生成的是一个 pickle 序列化文件，文件名称 p2pcse_cn.dict。新词词库的建立一方面通过人工录入，另一方面通过统计词汇出现的频率，比较普通词库，通过 Pickle 方法生成了系统中文分词新词词库，文件名称 p2pcse_en.dict。

4.2.3 中文分词算法

基于新词词库和普通词库，根据前文中文分词处理的设计目标，算法还必须处理歧义包容及准确性的问题。如果算法将所有有意义的分词单位都提取出来，可以很好的解决了歧义包含的问题，但新问题就出现了，假若某结点请求“中国人民银行”根据上面算法，将解析出有效分词单位为中国/中国人/中国人民/中国人民银行/人民银行/银行。虽然新词词库处理模块可以减少部分重复的搜索工作，但是在最坏的情况下，中文句子长度为 N 的字符串将解析出 $N(N-1)/2$ 个有效的分词单位，这导致分词的复杂度大大增加。而且很多组合对于请求者来说是毫无意义的。就像前文描述的，结点请求“中国人民银行”，分词单位中，“中国”，“中国人”，“中国人民”，“人民银行”，“银行”皆是不正确反映请求者的意图的。假如请求者的意图是“银行”那么可以肯定请求关键字将会为“银行”而非“中国人民银行”。

为解决准确性问题 P2PCSE 系统中文分词处理采取贪婪匹配法进行。就是说，某句子分词解析后，如果某分词单位包容于其他分词单位中，则取最大包容的分词单位。比如上面的“中国人民银行”最大分词包含单位为“中国人民银行”事实证明最大包容分词单位最能反映请求者意图，同时大大的降低了分词处理结果数量巨大的问题，同时提高了准确率，提高了分词系统模块的运行效率。

综合上述分析，首先将要解析的数据按标点符号打散成一个一个的句子，对句子进行分别解析，系统中文分词解析算法中文描述如下：

- 1) 利用 P2PCSE 系统首字哈希算法在 X-树初始节点 (FhashTable) 中查找句子首字指针所指汉字。如果查找成功记录 Fp 程序继续，如果查找失败，直接返回。
- 2) 根据 Fp 指针，在孩子节点(有序表)中查找后继汉字编码的位置，在查找成功的情况

下, 如果遇到 F_i 的值为 1 的结点, 就可以判定从首字到该结点的汉字组合为一个有意义的分词单位, 算法记录这个分词单位。如果 F_i 为 0, 那么将初始结点到该结点的组合根据 P2PCSE 新词词库哈希算法运算, 然后查找新词词库哈希表。如果该组合匹配新词哈希表中某地址, 记录该组合为有意义的分词单位。同时将首字指针移动到新词在原句子中的末尾位置。查找失败, 程序继续下一步。

- 3) 算法继续查找孩子结点如果 F_k 值为 1, 则可以判断第二步记录中的分词单位为冗余分词, 不能正确反映请求者意图, 删除该分词单位, 同时首字指针滑动到最新有意义分词单位在原句中末尾位置。如果 F_k 为 0, 第二步记录中分词单位保留, 重复第二步中查找成功 F_i 为 0 的步骤。算法重复 3) 的步骤直到孩子结点为空。
- 4) 待处理的句子的首字指针后移一位继续重复 1), 2), 3) 的步骤, 如此反复就可以将该待处理句子解析完成。

通过中文算法描述, 系统中文分词算法 CSECHSplitor 伪码描述如下:

```
void CSECHSplitor(sentence,*particArray)
{
    ....
    fisrtP = FHashTable.find(firstCharCode);
    while(fisrtP->child)
    {
        nextCharP = SFind(nextCharKey, fisrtP->child);
        if(nextCharP->Fi == 1)
        {
            del(previousParticUnit, particArray);
            add(currentParticUnit, particArray);
            fisrtP = getOriginalPos(currentParticUnit, sentence);
        }else{
            if(MatchNewWord(NewWordsHashTable, currentParticUnit))
            {
                add(currentParticUnit, particArray);
                fisrtP = getOriginalPos(currentParticUnit, sentence);
            }else{
                return false;
            }
        }
        fisrtP = FHashTable.find(fisrtP->charCode);
    }
    ....
}
```

从 X-树的结构及上述步骤我们可以看出, CSECHSplitor 查找算法每一次查找过程可以分为两步, 第一步是在首字哈希表中(FHashTable) 查找首字的过程, 时间复杂度为 $O(1)$, 第二步为在首字指针所指孩子节点的有序表中查找后继字及重复第二步的过程。据词库建立时候的统计数据显示, 孩子节点有序表的平均长度为 5。而首字哈希表所有子树的平均高度为 3。假如一个句子的长度为 N , 那么它的中文分词处理时间复杂度最好的情况下为 $O(N)$, 平均情况为 $O(C)C=16$, 在最坏的情况下, 时间复杂度也远小于 $O(N^2)$ 。

4.3 系统文件处理模块

P2PCSE 中文件处理模块负责索引解析共享文件夹下所有文件数据, 同时监控该文件夹处理新增或失效文件数据, 进一步与服务器超级节点数据同步。文件处理模块索引流程如下:

Step1) 首先对共享文件按照表 4-2 对文件各项参数进行提取。

Step2) 对需要进行中文分词处理的文件内容通过调用对应文件数据提取方法获取文件内容, 进一步调用分词模块处理分析。

Step3) 保存数据准备与超级节点进行索引数据同步。

表 4-2 文件处理归类表

文件类型	文件 大小	文件 名称	分词处理 文件名	分词处 理文件内容	文件数据提取方法
.txt	√	√	√	√	ParseTxt(filename)
.doc	√	√	√	√	ParseDoc(filename)
.htm,	√	√	√	√	ParseHtml(filename)
.pdf	√	√	√	X	
.vsd	√	√	√	X	
.pdf	√	√	√	X	
.rar	√	√	√	X	
.cmd	√	√	√	√	ParseCmd(filename)
.bat	√	√	√	√	ParseBat(filename)
.py	√	√	√	√	ParsePy(filename)
.mp3	√	√	√	√	ParseMp3(filename)

.wav	√	√	√	X	
.rm	√	√	√	X	
.rmvb	√	√	√	X	
.acc	√	√	√	X	
.avi	√	√	√	X	
.mp4	√	√	√	X	
.gif	√	√	√	X	
.jpg	√	√	√	X	
.png	√	√	√	X	
.psd	√	√	√	X	
.swf	√	√	√	X	
.flv	√	√	√	X	
.exe	√	√	√	X	
.ttf	√	√	√	X	
.dll	√	√	√	X	
.bash	√	√	√	√	ParseBsh(filename)
.xml	√	√	√	√	ParseXml(filename)
.asp	√	√	√	√	ParseAsp(filename)
.php	√	√	√	√	ParsePhp(filename)
无文件名	√	√	√	X	
因篇幅所限，本表未列举出所有扩展文件类型					

P2PCSE 文件处理模块对处理后的结果数据采用 XML 文档格式存储，其数据结构如表 4-3 所示。

表 4-3 P2PCSE 索引数据 XML 格式

索引数据 XML 格式示范
<pre> <?xml version="1.0" encoding="UTF-8"?> <P2PCSE_file> <filetype>filetype</filetype> <filesize>filesize</filesize> <filefullpath>filefullpath</filefullpath> </pre>

```

    <keyforfilename>keyforfilename</keyforfilename>

    <keysforcontent>key1 | key2 |...

    </ keysforcontent>

    <synchronizeflag>1|0</ synchronizeflag>

</ P2PCSE_file >

< P2PCSE_file >

    .....

    .....

    .....

</ P2PCSE_file >
    
```

各字段含义

P2PCSE_file	filetype	filesize	filefullpath	keyforfilename	keysforcontent	synchronizeflag
文件节点	文件类型	文件大小	文件全路径	文件名关键字	文件内容关键字	是否完成同步标志

系统文件处理模块只在普通节点上激活运行，利用 P2PCSE 普通节点的资源完成对共享文件的索引，从而减轻超级节点的压力。文件索引核心处理算法使用伪码描述如下：

Void ParseSharedFiles(SharedDirectory)

```

{
    For file in SharedDirectory:
        FileProperty=GetFileProperty(file); //对不同类型的文件按照表 4-2 规定的要求对文件名称和内容进行解析和分词处理。
        If(FileProperty)
        {
            NewFileXml =NewFileXml (); //生成新的空 XML 文档
            NewFileXml.writeFT(FileProperty.filetype); //写入文件类型
            NewFileXml.writeFFP(FileProperty.filefullpath); //写入文件全路径
            NewFileXml.writeFN(FileProperty.filename); //写入文件名称
            NewFileXml.writeFS(FileProperty.filesize); //写入文件大小
            NewFileXml.writeKFFN(FileProperty.keyforfilename); //写入经过分词处
        }
    }
}
    
```

```

    理的文件名称关键字
    NewFileXml.writeKFFC(FileProperty, keysforcontent); //写入经过分词处
    理的文件内容关键字
    NewFileXml.End();
}
else{
    Continue;
}
}
}

```

4.4 系统文件索引储存查询模块

P2PCSE 系统中文件索引储存查询模块是为 P2PCSE 系统提供共享文件全文检索数据储存, 节点查询搜索服务的核心模块。在整个系统中起关键的作用。普通节点关键字请求将通过分词模块处理分解为一个或者多个分词单位, 然后请求超级节点对该自治簇所有共享文件索引进行查询。综合目前储存解决方案, P2PCSE 系统文件索引储存模块可以采用如下方案:

- XML 文件储存方式
- 自定义二进制文件储存方式
- 关系数据库储存方式(以 Mysql 为代表)

通过综合数据测试分析对三种方案得出表 4-3 所示:

表 4-3 P2PCSE 索引存储解决方案比较

存储类型	查询需要转化	线程安全	查询速度	数据大小限制	并发支持
XML 文件	否	否	慢	无限制	不支持
自定义二进制	是	否	慢	无限制	不支持
关系数据库	否	是	快	大于 100G	支持

考虑到 P2PCSE 网络模型对超级节点及引导节点可能的最大并发请求, 根据表 5-1, 以 100W 节点规模 P2PCSE 网络, 平均每个超级节点维护 2000 普通节点为例。该网络将有 500 个超级节点, 在最坏的情况下, 某超级节点将接受 22000 次并发搜索请求。这就要求索引存储查询方案必须支持大规模并发, 而且查询速度快的。

通过上面综合分析, P2PCSE 系统采用开源的关系数据库 MYSQL 方式存储自治簇共享文

件索引数据数据。

在 P2PCSE 系统中普通节点综合分词系统及文件属性处理系统索引本机共享文件，在初始化和更新的情况下发送索引数据给超级节点。

超级节点只负责存储自治簇普通节点的经过分词处理后的关键字索引，超级节点本身不执行分词处理过程。其中负责存储对应自治簇共享文件索引数据的核心表 *Autonomy_Cluster_Para* 如表 4-4 所示, *Autonomy_Cluster_Key* 如表 4-5 所示。

表 4-4 共享文件参数储存表

File_id	Peer_id	File_type	File_size	File_fullPath	File_nameKey
文件 ID, 表主键	所在普 通节点 ID	文件类型	文件大小	文件全路径	文件名经分词处 理后的关键字列 表

表 4-5 共享文件内容关键字储存表

Key_id	File_id	Peer_id	Key_list
表主键	文件 ID, 外键	普通节点 ID, 外键	分词单位列表用“,”分隔

4.5 本章小结

在第三章 P2PCSE 系统设计方案分析的基础上对系统关键——全文检索的实现做了详细分析。综合分析了全文检索子系统的模型，提出了一种基于 X-树的中文分词算法 CSECHSplitor，对其模型及算法进行了详细的定义和实现。对全文检索子系统按照功能划分为分词处理模块、文件处理模块、索引储存查询模块并分别对其进行详细介绍和实现。

第五章 P2PCSE 系统网络结构实现

第三章对 P2PCSE 系统总体设计做了详细的分析, 其中对 P2PCSE 系统的关键部分——网络结构做了分析, 为本章重点分析并实现系统网络结构做了铺垫。本章开篇首先介绍 P2PCSE 系统网络结构模型, 将网络中节点按照功能划分为引导节点、超级节点、普通节点并分别进行定义, 对节点间网络连接模型进行分析。在此基础上对实现 P2PCSE 系统网络结构的关键部分分别进行介绍和实现。

5.1 系统网络模型

P2PCSE 系统网络结构基于改进的三层混合式半分布网络, 其模型结构如图 5-1 所示, 网络中节点根据节点硬件及网络性能分为普通节点[NNode], 超级节点[SNode], 同时另外设置一个引导节点[GNode]。

其中引导节点及超级节点都为一台或者多台(最多 10 台)普通 PC 或者服务器构成的网络, 其模型结构如图 5-2 所示, 子网络节点间采用全分布非结构化连接。实行互相索引备份, 增强系统健壮性, 防止某点出现故障时候导致整个或者局部网络失效, 对搜索请求进行动态分配。

引导节点与超级节点间采用中心化拓扑方式连接。引导节点负责新节点加入时候动态分配超级节点, 超级节点的维护, 超级节点失效时重新分配超级节点给普通节点。

超级节点与附近的一定规模的普通节点构成一个自治的节点簇, 簇内采用传统的集中目录式结构, 其模型结构如图 5-3 所示, 超级节点负责储存索引该簇内所有普通节点数据索引, 提供数据查询, 信息中转服务, P2PCSE 网络中超级节点间采用纯 P2P 网络结构连接。

普通节点提供 P2PCSE 系统网络共享文件来源, 负责对各自节点共享文件数据进行检索, 并与其所在的自治簇超级节点进行索引数据同步。

在 P2PCSE 系统中, 为了避免超级节点间搜索请求转发造成网络数据堵塞及最坏情况下某超级节点超出极限值, 规定 P2PCSE 系统中每个超级节点最多接受 3000 个并发搜索请求, 超出限制返回服务不可用标志, 直接拒绝请求。

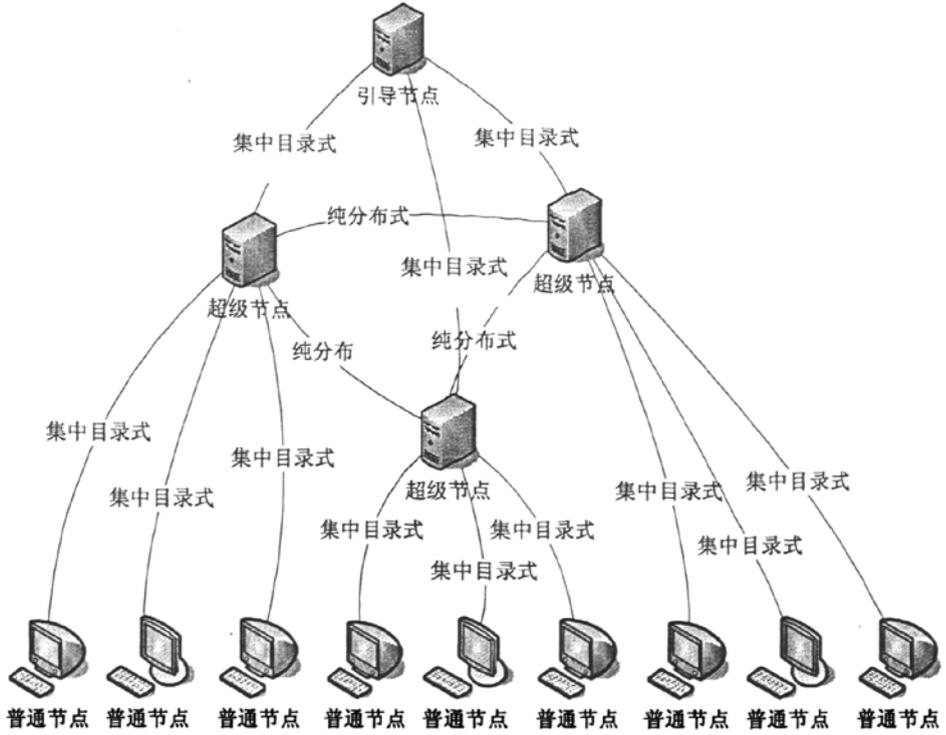


图 5-1 P2PCSE 改进的三层混合网络结构模型

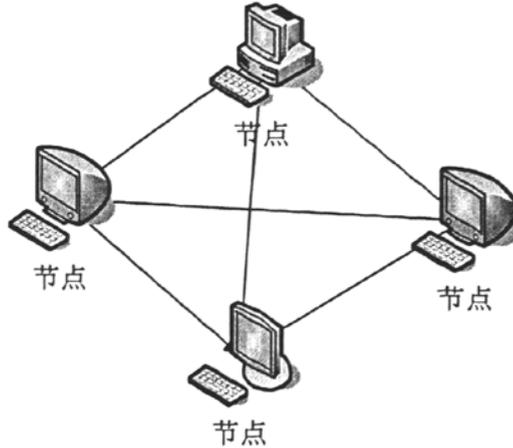


图 5-2 P2PCSE 系统中超级节点及引导节点结构

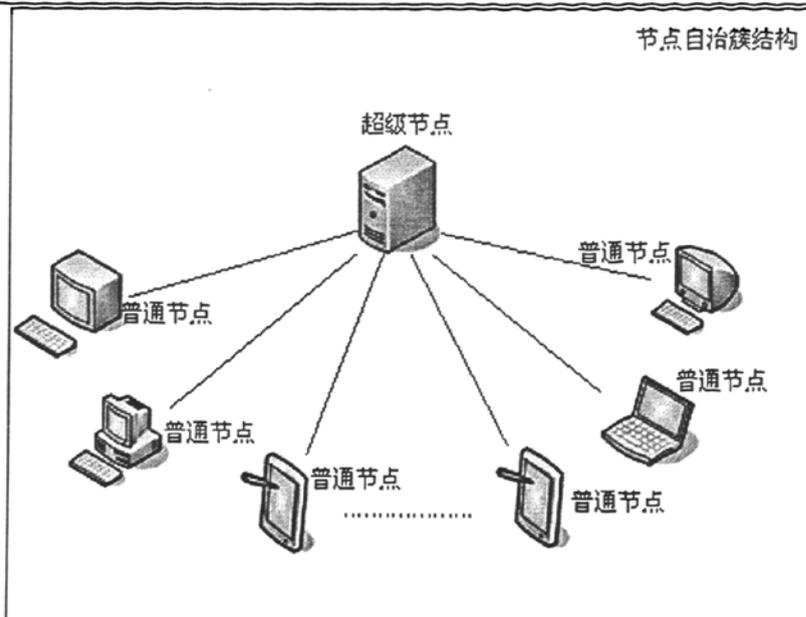


图 5-3 P2PCSE 系统自治簇结构模型

P2PCSE 两层混合网络模型中，在 10 万规模普通节点的网络中，经过统计节点共享文件总数平均在 3000 个以内，每个文件经过 P2PCSE 索引模块处理后的索引数据约为 0.5K[见下文]，每个节点的索引数据平均在 $3000 \times 0.5 = 1500K$ (1.5M)，假如每个超级节点负责维护 1000 个普通节点的文件索引数据，每个超级节点负责维护的索引数据总量将在 $1.5M \times 1000 = 1.5G$ 左右。那么我们可以计算出超级节点的个数为 $10,000/1000 = 10$ 个。

从下表我们可以看出不同规模网络的所需超级节点总数及索引总数据。

表 5-1 不同规模网络超级节点数目及索引数据分析

类型/规模	10 万	100 万	1000 万	5000 万
方案一：每个超级节点维护 500 个普通节点				
超级节点数目	200	2000	20,000	100,000
索引数据大小	750M	750M	750M	750M
方案二：每个超级节点维护 1000 个普通节点				
超级节点数目	100	1000	10,000	100,000
索引数据大小	1.5G	1.5G	1.5G	1.5G
方案三：每个超级节点维护 2000 个普通节点				
超级节点数目	50	500	5000	50,000
索引数据大小	3G	3G	3G	3G
方案四：每个超级节点维护 5000 个普通节点				

超级节点数目	20	200	2000	20,000
索引数据大小	7.5G	7.5G	7.5G	7.5G
方案五：每个超级节点维护 10000 个普通节点				
超级节点数目	10	100	1000	10,000
索引数据大小	15G	15G	15G	15G

从上表看出每个节点维护大于等于 2000 个普通节点的对超级节点的索引存储压力是适中和可取的。在 P2PCSE 系统中，采用表 3-1 中的方案三。每个超级节点维护 2000 个普通节点。在 P2PCSE 系统中三层混合网络结构中每个超级节点负责维护 2000 个普通节点，每个超级节点最大索引数据为 3G。

5.2 P2PCSE 系统网络结构实现

在 P2PCSE 系统设计方案中对网络结构分析的基础上，本节对实现 P2PCSE 网络结构模型的关键处理方法进行了详细分析及实现，下面将一一介绍关键处理办法细节。

5.2.1 对超级节点产生的处理

超级节点是对应自治簇的核心，在 P2PCSE 中通过特定条件选择超级节点。普通节点在加入网络前，先计算自身硬件及网络性能是否满足超级节点的性能要求，然后提示用户该节点适合成为超级节点。再得到用户许可后，发送申请成为超级节点的请求 REQUESTSNODE<DATA>，DATA 参数包含节点的网络地址，性能评估，系统引导节点接受到该信息后，经过综合评估，对符合成为超级节点的请求予以成功回复 REPLY<DATA>，否则拒绝 ERROR<DATA>，该普通节点接受到反馈信息后，根据信息类别设置自身为超级节点或者设置为超级节点子节点，或者重新申请加入网络。

该过程用程序伪码描述为：

```
Void RequestSNode(DATA)
{
    //判断节点综合性能是否满足要求
    Result=PeerFitForSNodeByPerformance();
    If(Result>MinStand)
    {
        //申请成为超级节点
```

```
Response=SendDataTo(Nnode,DATA);
If(Response==REPLY)
{
    //设置为超级节点, 或者为超级节点子节点
    SetAsSNode();
    //显示超级节点相关信息
    ShowSNodeInfo();
}else{
    //显示不符合原因
    ShowErrorInfo();
    //申请加入网络
    JOIN(PEERID);
}
}else{
    //申请加入网络
    JOIN(PEERID);
}
}
```

5.2.2 对新节点加入的处理

和现有的 DHT 一样, P2PCSE 网络中新节点的加入至少要知道一个活动节点的地址 (IP, 端口), 而这个已知节点在本系统中设置为系统引导节点。

新节点加入时, 首先向系统引导节点发送连接请求 JOIN<DATA>, 引导节点收到请求后根据已经维护的超级节点列表选取合适的超级节点返回 REPLY<DATA>, 其中 DATA 包括超级节点地址, 否则引导节点 [NNode] 因为节点已经登陆或其他不适合登陆原因拒绝新节点请求, 返回给新节点 ERROR<DATA>, DATA 包含具体登陆失败信息。

当新节点成功收到引导节点的 REPLY<DATA> 信息后, 根据 DATA 参数中提供的超级节点地址信息向该超级节点发送 JOIN<DATA>, DATA 参数包含自身地址参数, 超级节点接受连接返回 REPLY<DATA>, DATA 参数包含欢迎及该自治簇相关信息。

新节点成功加入网络。否则收到 ERROR<DATA> 数据, 申请加入网络失败, 其流程如图 5-4

所示。

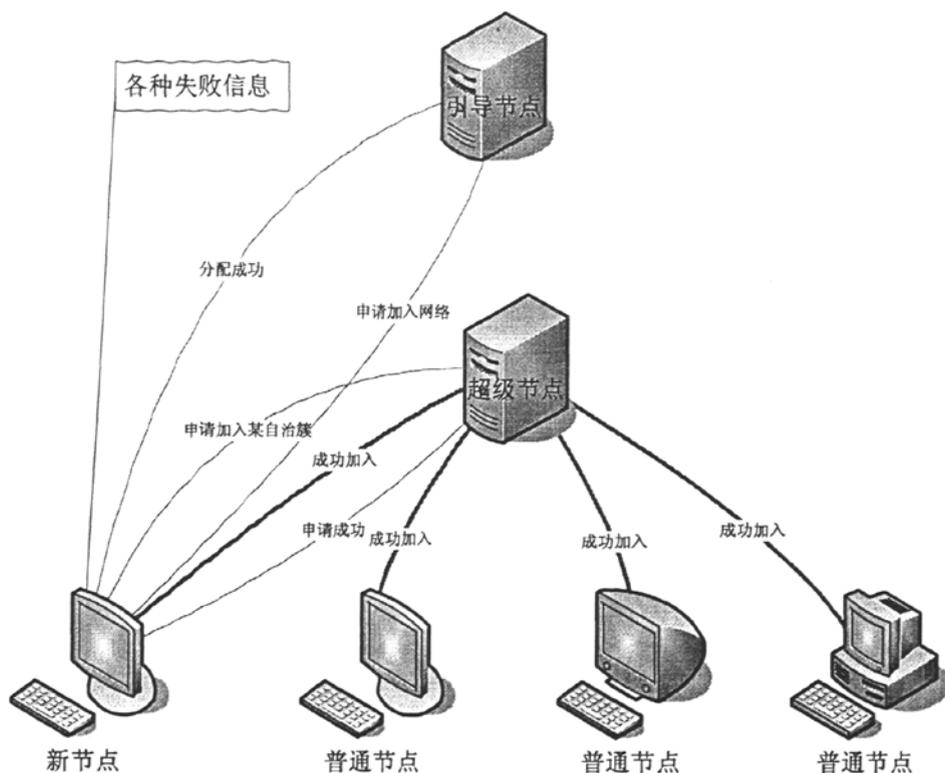


图 5-4 新节点加入流程图

5.2.3 对普通节点失效和退出的处理

在用户关闭 P2PCSE 程序时候，程序向该节点所属自治簇中超级节点发送 PEERQUIT<DATA>请求，对应超级节点收到该请求后，将 DATA 参数所代表的节点中自身节点列表中删除，并返回 REPLY<DATA>数据，程序收到该信息后，正常退出。

由于网络中断等原因，在客户端还未来得及向对应超级节点发送退出请求时，该节点在此情况下属于失效节点，在 P2PCSE 系统中，存在一种判断节点是否有效机制，在自治簇中，超级节点每 30 分钟采用多线程方式向自身所维护的普通节点列表发送 PING<DATA>数据，目标节点接受到该命令后，如果程序正常运行立即返回 ALIVE<DATA>数据表明节点正常运行。如果节点超时没有返回，超级节点判断该节点失效，同时将其从自身维护的节点列表中删除。

正常退出情况用程序伪码描述为：

```
Void PeerQuit(PEERID)
```

```

{
    SendDataTo(Snode,PEERQUIT);
    PeerQuitDone();
}

```

失效情况下超级节点判断节点是否有效其伪码描述如下:

```

Void ChkPeersAlive(PEERLIST)
{
    For Peer in PEERLIST:
        New Thread(ChkAlive,PEER);
}
Void ChkAlive(PEER)
{
    Try
    {
        Response=SendDataTo(PEER,PING,DATA);
        If(Response)
        {
            DoNothing();
        }else{
            DeleteExpiredPeer(PEER);
        }
    }Exception(TimeOut e){
        DeleteExpiredPeer(PEER);
    }
}
}

```

5.2.4 普通节点文件索引的实现

当节点成功加入网络后,节点首先遍历共享文件夹,利用 P2PCSE 全文检索模块索引归类所有共享文件,进而与超级节点进行数据同步。普通节点与自治簇超级节点文件索引信息交换包括两种类型:

- 文件增加或者更新

当普通节点初始化时候，或者普通节点共享文件夹中增加了新的文件或者文件内容改变，为了让自治簇超级节点保持对整个自治簇节点共享文件索引的完整及一致性，该普通节点需要提交增加新文件或者更新文件的请求，该普通节点向对应超级节点发送 UPDATEFILE<DATA>，DATA 参数包括该普通节点网络 ID，更新或者增加动作及更新或者增加的文件经过 P2PCSE 索引模块处理后的结果。超级节点接受该参数后根据网络 ID，更新或者增加动作，进行相应更新数据或者增加数据操作，完成后返回 REPLY<DATA>成功信息，该普通节点标志该记录成功。否则由于超时或者网络中断原因该普通节点记录该信息为未完成状态，等待 10 分钟后重新重复以上过程。

该过程程序伪码描述如下：

```
Void UpdateFile(DATA)
{
    Response=SendDataTo(Snode, DATA);
    If(Response)
    {
        MarkSuccess4LocalFileUpdate(DATA);
    }else{
        MarkFailFlag4LocalFileUpdate(DATA);
        TimeDelay();
        UpdateFile(DATA);
    }
}
```

- 文件删除或者失效

当用户删除或者移动共享文件夹中某文件时候，自治簇超级节点需要维护该簇网络共享文件索引的一致。该普通节点需要通告对应超级节点某文件已失效，发送 DELETEFILE<DATA>，DATA 参数包括普通节点网络 ID，失效文件标志信息。对应超级节点接受该参数后，找到对应记录后删除，返回 REPLY<DATA>成功信息，该普通节点删除对应记录，过程完成，否则记录该操作未完成，等待一 10 分钟后重复以上过程。

该过程程序伪码描述如下：

```

Void DeleteFile(DATA)
{
    Response=SendDataTo(Snode, DATA);
    If(Response)
    {
        MarkSuccess4LocalFileDelete(DATA);
    }else{
        MarkFailFlag4LocalFileDelete (DATA);
        TimeDelay();
        DeleteFile (DATA);
    }
}

```

5.2.5 对文件搜索请求的处理

当某普通节点发出请求时，请求将先发送到自治簇的超级节点 QUERY<DATA>,DATA 包含请求节点网络 ID<PEERID>,搜索关键字<KEY>,最少文件数目限定<MINRECSET>,搜索超时限定值<STIMEOUT>。超级节点搜索符合该关键字语义的文件记录，如果结果未满足请求节点设置的最少文件总数，即搜索不充分的情况下，该超级节点根据请求节点设置的最大转发限定值随机转发搜索请求给 P2PCSE 网络中其他超级节点，其他超级节点搜索到符合请求的记录后根据 DATA 提供的请求节点网络 ID，将结果发送给请求节点 QUERYRESPONSE<DATA>,DATA 参数包含目标文件信息，所处目标节点网络信息。请求节点接受搜索结果信息后将数据显示，以供搜索者抉择。

该过程程序伪码描述如下：

对超级节点端

```

Void Search(DATA)
{
    //超级节点搜索数据库所储存的文件索引记录
    RESPONSE=P2PCSE_SEARCHER(DATA.KEYINFO);
    If(ChkResultFitRequest(RESPONSE, DATA . MINRECSET))

```

```
{
    SendDataTo(DATA.PEERID,RESPONSE);
}else{
    //在搜索不充分的情况下转发搜索请求
    RandSendDataTo(SnodeList,DATA);
}
}
对请求节点端
Void Query(DATA)
{
    SendDataTo(Snode,DATA);
    RESULT=QueryReceiver(DATA. STIMEOUT);
    If (RESULT)
    {
        ShowQueryResult(RESULT);
    }else{
        ShowErrorInfo();
    }
}
```

5.3 本章小结

本章开篇系统的介绍了 P2PCSE 系统使用的三层混合式网络结构模型,并将节点按类别划为普通节点、超级节点及引导节点,对 P2PCSE 系统网络结构各节点功能进行了详细定义。接着对实现 P2PCSE 系统网络的各种主要功能进行分析和设计,并给出了伪码实现。本章和第四章是 P2PCSE 系统的关键部分,为第六章 P2PCSE 原型系统的实现打下了基础。

第六章 P2PCSE 原型系统的实现与性能测试

在第三章对 P2PCSE 系统总体设计、第四章对全文检索子系统,第五章对网络结构等关键技术实现的基础上,采用 Python 语言实现了 P2PCSE 方案的原型系统,本章主要对 P2PCSE 系统文件搜索的流程、用户界面进行详细描述,详细测试了 P2PCSE 系统中文分词处理模块并与常见中文分词系统进行对比,最后对 P2PCSE 系统网络结构各方面性能进行测试,得出 P2PCSE 系统性能总结。

系统具体开发环境:

开发环境:windows xp sp3, cpu:intel Pentium Dual T2310 1.46GHZ,2G DDR

开发相关:Python2.5, Wxpython2.8(界面语言),Mysql(数据库)

开发工具:Eclipse

系统运行平台:windows,unix,linux,mac os 等操作系统。

6.1 P2PCSE 系统文件搜索流程

在 P2PCSE 系统中普通节点初次运行首先设置共享目录,然后利用 P2PCSE 文件索引模块全面索引本地共享文件,将其索引数据与该节点所在自治簇超级节点同步。普通节点执行搜索请求,搜索请求将发送到该自治簇中的超级节点,超级节点查询数据库中储存索引数据表,返回请求结果,在结果总数不符合请求者要求的情况下,超级节点将请求随机转发个制定数目的其他超级节点。超级节点再搜索到数据后,根据请求信息将搜索结果返回给请求节点,整个搜索过程完成。其具体流程如图 6-1 所示。

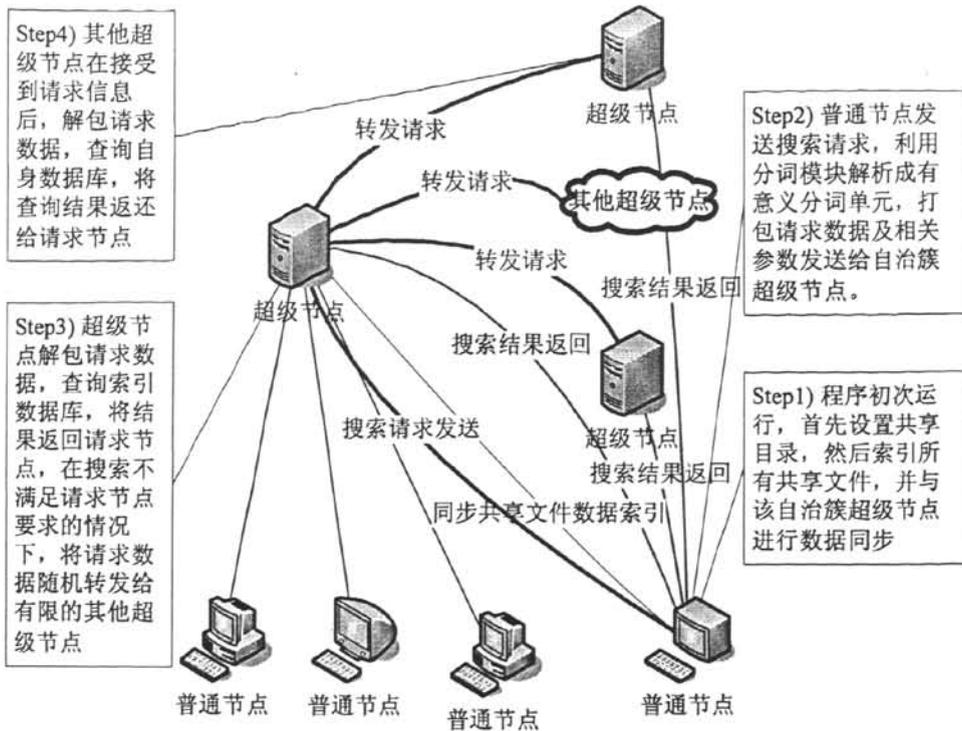


图 6-1 P2PCSE 系统文件搜索流程图

6.2 系统用户界面实现

友好的用户界面给用户带来良好的体验, P2PCSE 界面设计具体包括: 菜单部分, LOGO 部分, 搜索输入框, 功能按钮部分, 及列表显示部分。从而用使用者可以通过 P2PCSE 程序, 方便的进行系统设置、搜索数据、发送信息、预览搜索结果、下载文件等。其中设置菜单可以让用户设置搜索结果总数范围、共享目录、及一些其他参数。P2PCSE 设置共享文件目录如图 6-2 所示, 节点加入网络后界面如图 6-3 所示, 节点发送信息界面如图 6-4 所示, 节点接受到信息后如图 6-5 所示, 节点搜索文件后结果列表如图 6-6 所示。

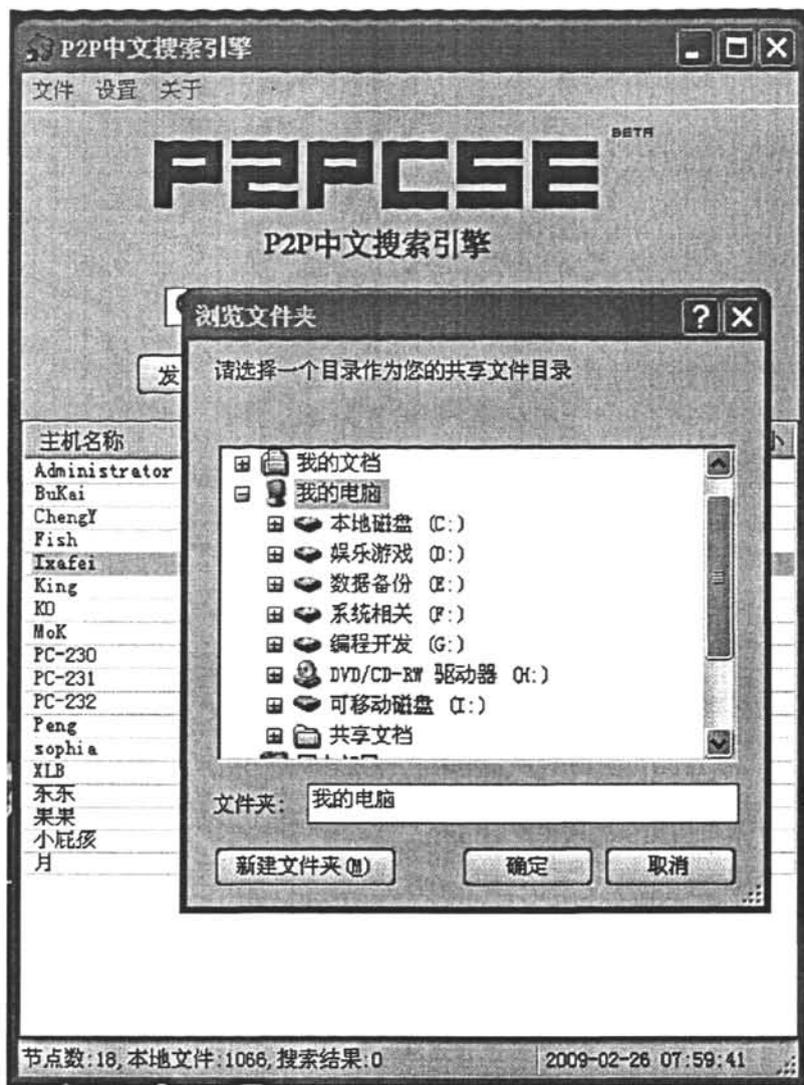


图 6-2 P2PCSE 用户设置共享文件目录

当用户运行 P2PCSE 系统时，如果是初次运行，程序要求设置共享目录，用户也可以在程序运行过程中点击设置更改共享目录地址。如图 6-2 所示，节点在成功加入网络后得到本簇节点列表，共享目录文件总数 1066 个，因为用户没有开始搜索，搜索结果为 0。

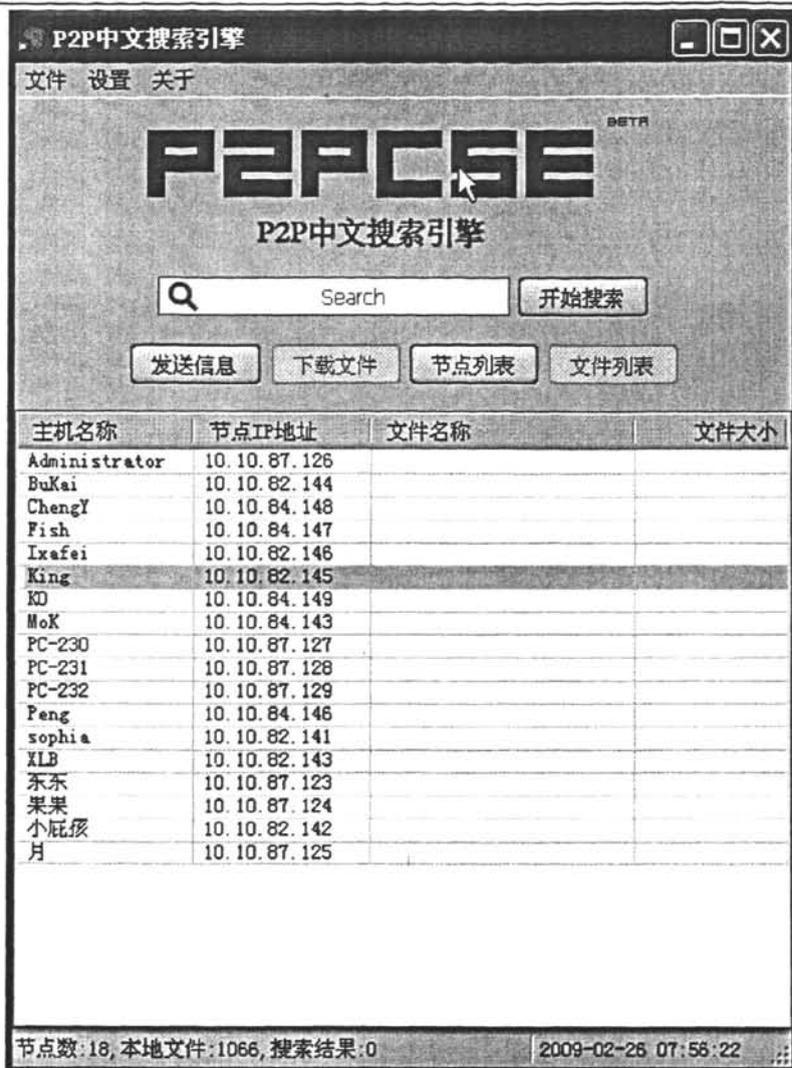


图 6-3 P2PCSE 系统节点加入网络后节点列表

如图 6-3 所示，节点在成功加入网络后获得该自治簇所有节点信息。节点信息包括两个部分：主机名称，节点 IP 地址。同时状态栏显示节点总数为 18 个。

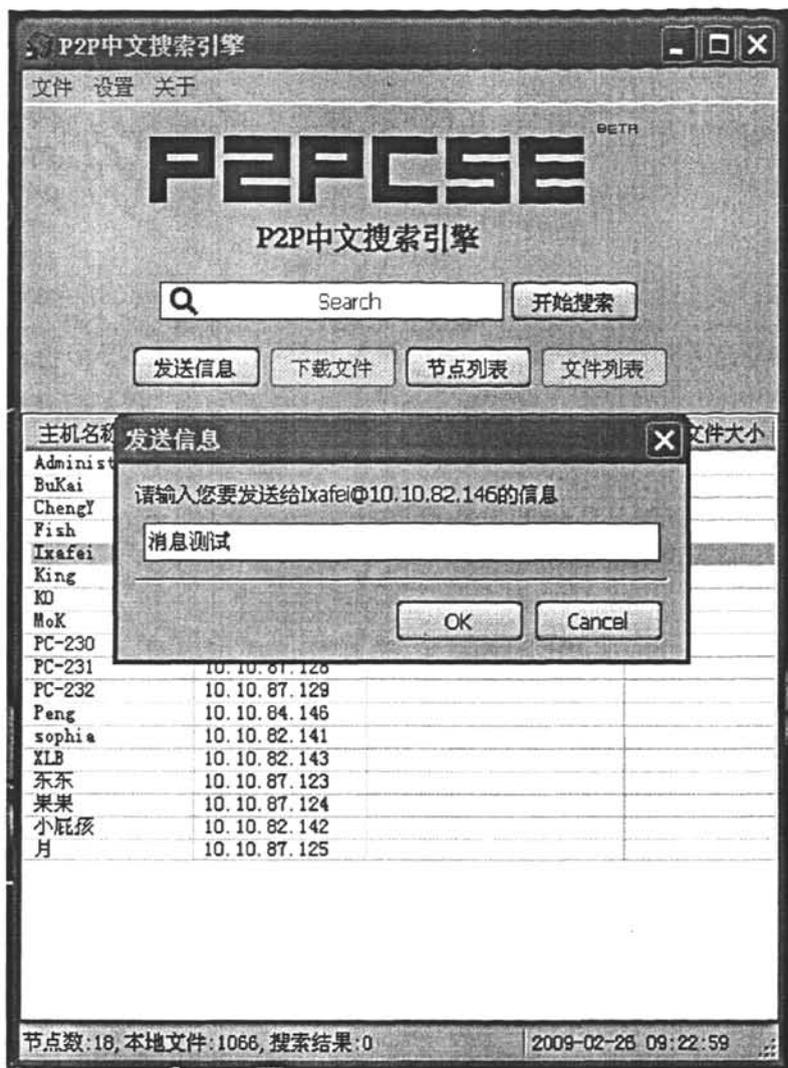


图 6-4 P2PCSE 系统节点发送信息界面

P2PCSE 系统允许节点间发送简单文字信息，如图 6-4 所示节点 sophia 选定主机名称为 Ixafei 的节点，点击发送信息，出现信息发送具体界面。点击 OK 按钮可以将信息发送出去，点击 Cancele 按钮将取消信息发送。

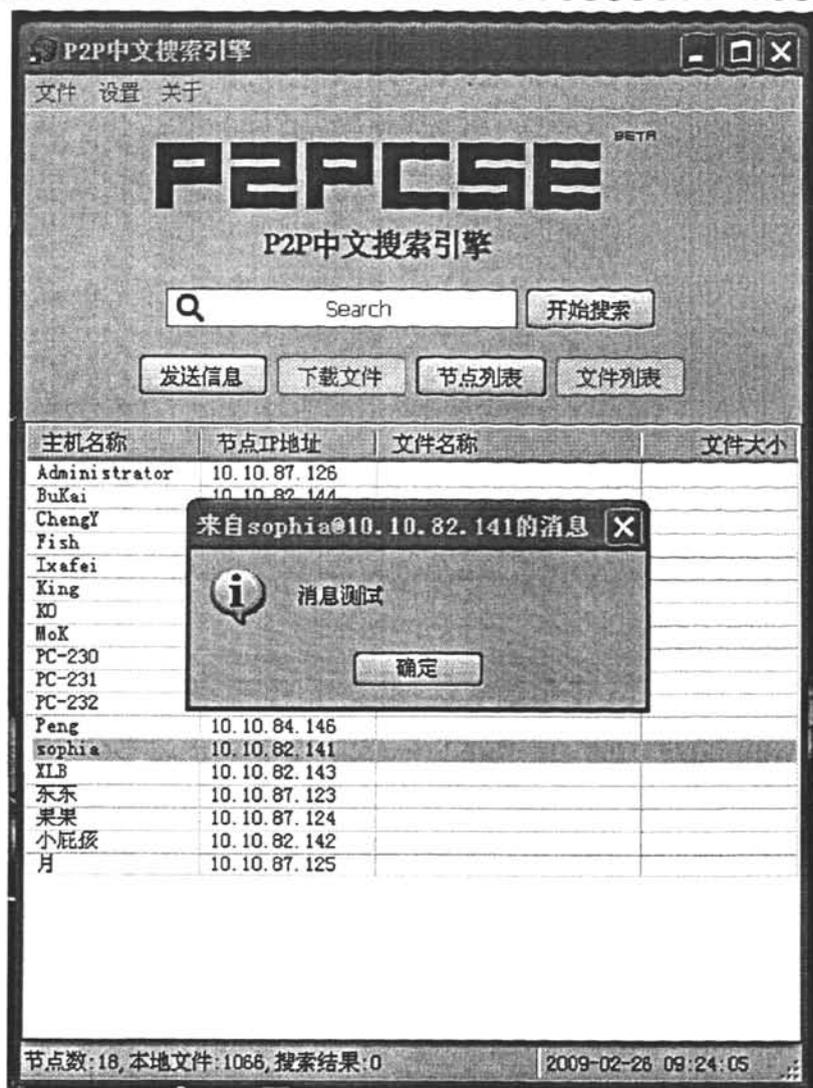


图 6-5 P2PCSE 系统节点接受到信息后界面

节点接受到其他节点的文字信息如图 6-5 所示，信息将由弹出提示框显示，提示框标题为发送信息来源节点的主机名称和主机 IP 地址，点击确定按钮可以关闭信息。

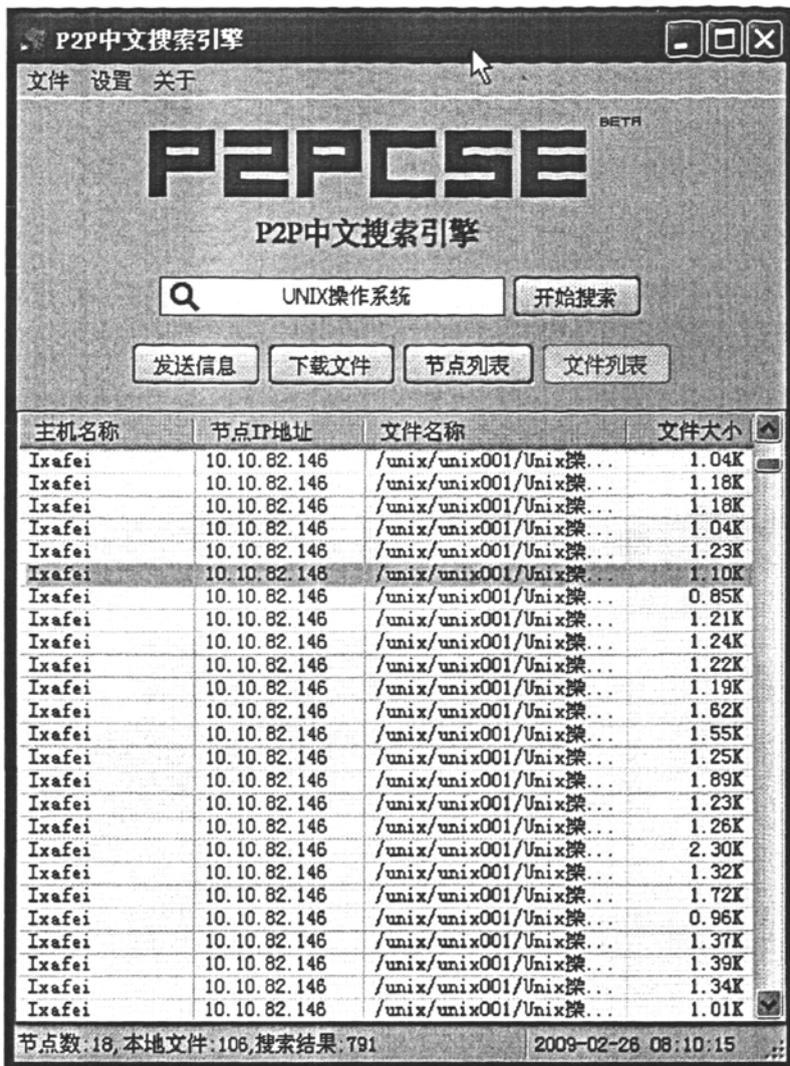


图 6-6 P2PCSE 系统节点搜索结果界面

如图 6-6 所示节点发送搜索“UNIX 操作系统”的关键字请求，结果成功返回后程序界面自动由节点列表切换为文件列表。状态栏显示该自治簇节点总数为 18 个，本地共享文件 106 个，搜索后返回 791 个符合“UNIX 操作系统”请求的结果。

6.3 P2PCSE 全文检索子系统性能测试

6.3.1 中文分词处理模块性能测试

在实验中我们选了多篇平均长度为 5004 字(包括空格)的中文新闻内容。然后用四种流行分词处理办法及 CSECHSplitor 对它进行处理，来检验基于 X-树的分词处理性能，实验

结果如图 6-7 所示,其中“|”仅为了读者容易分辨分词单位所特加,经过整理得出统计数据如表 6-1 所示:

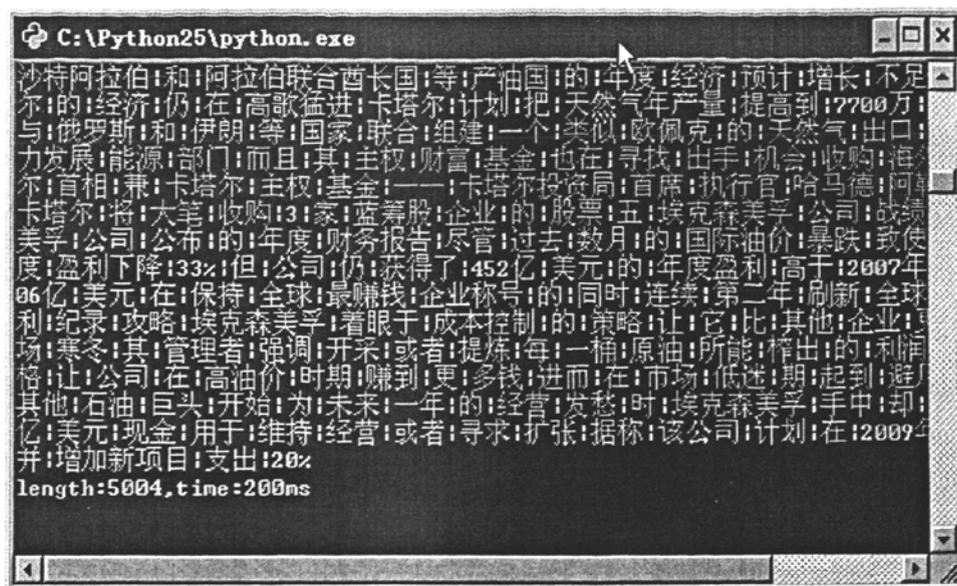


图 6-7 CSECHSplitor 算法分词处理结果图

表 6-1 CSECHSplitor 分词法与常见中文分词法性能比较

算法名称	算法描述	平均耗时	平均准确度
简单分词法	将文章按标点符号隔开成句子	460ms	无
标准分词法	将文章的中文字分成一个一个的单字	500ms	<30%
最大匹配法	先加载词库,将文章中的文字分成词和字	540ms	<80%
海量分词法	将文章中的文字分成词和字	220ms	95%
CSECHSplitor 分词法	先加载词库,基于 X-树结构将文章中的文字分成词和字	200ms	约 93%

实验结果表明:最大匹配法分词效率很低,而且不准确,歧义包容处理较差,海量分词[56]是国内做的非常好的专业化分词工具,其效率与准确率高;特别当文章中包含较多的新词时候,标准分词法和最大匹配法表现极差,准确度分别为 0 和小于 30%.当使用 CSECHSplitor 分词法的时候,搜索 X-树结构,及采用普通词库和分词词库分开处理,分词效率较高,准确率也较高。

6.3.2 全文检索子系统准确度与满意度测试

使用 PC 机中的两作为超级节点, 另外五台作为普通节点构建两个自治簇。经过所有普通节点完成索引数据解析及与超级节点同步后, 开始分别测试 P2PCSE 系统搜索准确度和满意度, 其中准确度 (C) 定义为搜索结果数据中符合搜索关键字的记录条数 (F) 与结果总记录条数的比例 (T), $C=F/T*100\%$ 。满意度 (S) 定义为在网络中实际符合请求者记录条数 (I) 大于等于请求者要求满意结果条数 (U) 的前提下, 请求者得到的满意结果条数 (R) 与要求满意结果的条数 (U) 百分比, $S=C*R/U*100\%$, $I \geq U$ 。

分别用表 6-2 中新词关键字, 普通关键字各 10 个进行测试, 并设置文件 U 等于 I, 统计平均结果如表 6-3 所示:

表 6-2 测试用关键字表

新词关键字	稀饭 果酱 走召弓虽 咔嚓 酱紫 8 错 饭饭 片片 斑竹 水桶 潜水员 恐龙 青蛙 小强 抓狂 包子 小虾 老鸟 大虾 爱老虎油
普通关键字	时候 敏锐 发现 房地产业 存在 泡沫 经济 滑落 费尽周折 说服 投资者 相信 美国 房地产 市场 危机 保护 获利 一本万利 疑虑

表 6-3 P2PCSE 原型系统与相关系统综合性能对比

关键字类别	准确度	满意度
普通关键字	95%	95%
新词关键字	95%	90%
对比采用 Chord 模型的普通 P2P 搜索系统[57]		
普通关键字	67%	8%
新词关键字	83%	12%

上述结果表明 P2PCSE 系统的搜索准确度及满意度是很高的, 充分证明了 P2PCSE 系统自定义的全文检索子系统模型及 CSECHSplitor 算法的正确、可行性。之所以准确度和满意度未能达到 100%, 关键原因在于普通词库和新词词库的收录数量。理想状态下普通词库及新词词库收录 100% 的数据, 理论上是可以准确度和满意度达到完美的结果。

6.4 P2PCSE 网络结构性能及压力测试

测试环境: 教研室 PC 机若干台, CPU: 英特尔奔腾酷睿双核 T2310 1.46GHZ 硬盘: WD 120G IDE 内存: DDR2 2G 系统: windows xp sp3 或 Ubuntu 8.10 网络: 局域网 1000M Hub。

采用四台安装有 Ubuntu 操作系统的普通 PC 中的两台组建 P2PCSE 系统引导节点, 另外两台组建一个超级节点, 在另外三台安装有 Ubuntu 操作系统的普通 PC 作为测试节点构建 P2PCSE 网络, 令节点性能压力 $P = (\text{内存占用量} + \text{CPU 占用量}) / 2 * 100\%$, 测试机器在无负载的情况下 P 平均为 20%。根据不同条件请求下得出不同 P 的取值, 绘制出性能测试图。

6.4.1 系统在新节点加入时性能分析

利用 Python 程序模拟新节点的加入不同并发数目时请求对系统引导节点的压力进行分析。Python 模拟程序分别模拟了 0 到 30,000 个节点同时并发发送加入网络的请求给引导节点及超级节点, 同时随机让组成超级节点和引导节点的一台 PC 断开网络。得出统计数据如图 6-8 所示:

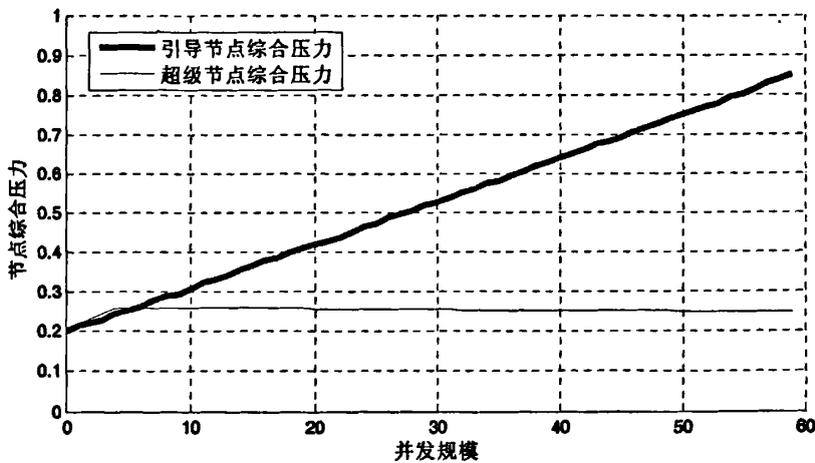


图 6-8 新节点加入时节点综合压力测试

其中 Y 轴代表节点的节点性能压力(参见 6.4 节), X 轴代表并发请求规模, 单位为 500。由图 6-8 知新节点加入先对引导节点发起请求, 引导节点全局配置超级节点信息, 根据引导节点返回信息, 新节点向超级节点发送加入请求并握手, 该超级节点首先执行判断该节点是否属于新节点还是以前连接成功后来因为网络中断、主动下线等原因造成的重新连接。在该超级节点剩余维护能力范围内, 向请求节点发送准许连接标志等系列动作。在 P2PCSE 系统中一个超级节点只负责维护 2000 个普通节点, 所以当节点超过 2000 后, 系统引导节点不再增加新节点给该超级节点, 超级节点性能保持为常数。在相同规模的新节点请求加入的过程中, 超级节点比系统引导节点多执行一些相关操作, 故在 0 到 2000 的节点规模, 超级节点压力明显大于系统引导节点, 系统测试结果与理论分析保持一致。

通过上面的分析可以看出 P2PCSE 模型使用的网络结构对超级节点及引导节点性能有

好的平衡保护作用，即使并发规模很大，也不会造成引导节点和超级节点的崩溃，在组成超级节点和引导节点的一台 PC 断开网络的情况下，系统依然保持稳定，相比传统半分布式网络，P2PCSE 系统更易扩展、更稳定同时超级节点及引导节点更加可靠。

6.4.2 系统在节点失效和退出时性能分析

利用 Python 程序模拟 2000 个普通节点首先加入网络与超级节点构建成为一个自治簇。然后 Python 程序模拟节点同时失效数目从 0 到 2000，设置超级节点失效判断机制时间间隔与该 Python 模拟程序同步。经过综合数据得出结果如图 6-9 所示：

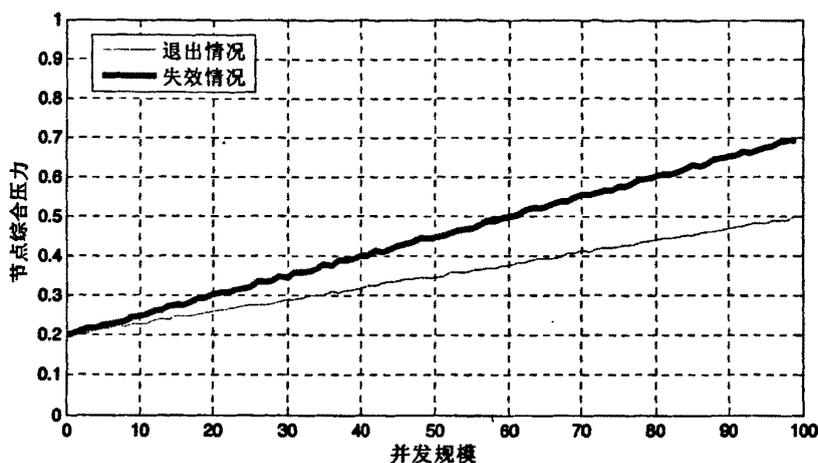


图 6-9 失效或退出时候节点压力综合测试

其中 Y 轴代表节点的节点性能综合压力，X 轴代表并发请求规模，单位为 20。从图 6-9 可以看出，相同数目的普通节点在失效的情况下超级节点所占的性能明显高于普通节点退出的情况，是因为普通节点失效将由超级节点主动发送基于 UDP 通讯的 PING 请求然后等待返回结果或者超时结果，而普通节点退出则是超级节点被动接受普通节点的退出请求，超级节点所执行的主动请求动作对机器性能消耗比被动接受明显要多很多。

通过上面综合分析及图 6-9 可以看出，P2PCSE 系统在节点失效和退出最坏的情况下，节点依然在一个安全范围，可见 P2PCSE 系统具备良好的可靠性。

6.4.3 系统在普通节点同步文件索引时性能分析

利用 Python 程序模拟从 0 到 2000 个普通节点并发情况同时发送 P2PCSE 系统所规定的文件索引 XML 文件，单个索引 XML 文件大小为 1K 情况下对自治簇超级节点及 P2PCSE 系统

引导节点的性能压力，得出统计数据如图 6-10 所示：

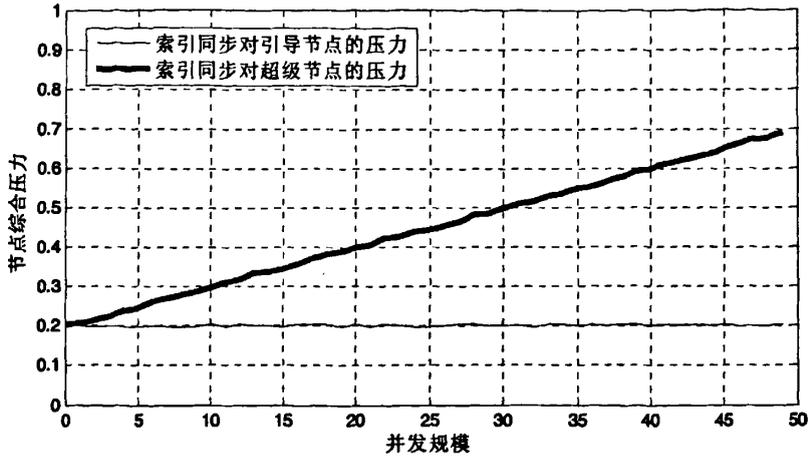


图 6-10 索引同步时节点压力综合测试

其中 Y 轴代表节点的节点性能综合压力, X 轴代表并发请求规模, 单位为 40。由图 6-10 可见, 在 P2PCSE 系统中文件索引数据同步对系统引导节点并未造成压力, 因为自治簇节点同步索引数据时与系统引导节点并无交互, 所以系统引导节点性能保持平稳。而超级节点需要执行接受请求和解析 XML 文件并保存进入数据的请求, 随着并发的增大, 超级节点所受压力增加。

从图 6-10 可以看出, 即使自治簇内执行索引同步操作的节点达到极限值, 超级节点压力保持在 70% 以内。P2PCSE 系统并不因特殊情况而导致局部崩溃。

6.4.4 系统在文件搜索请求时性能分析

我们使用 PC 机中的一台, 安装 Ubuntu 操作系统作为超级节点, 使用 Python 程序实现一个模拟测试节点发送搜索请求, 退出请求。因为超级节点间转发请求和普通节点对超级节点的请求实际是一样的, 所以这个模拟节点可以模拟了 10W 节点规模的网络到 100W 节点规模的网络对每个超级节点的搜索性能。P2PCSE 系统设计方案在 10W 节点规模的网络, 平均每个超级节点维护 2000 个普通节点, 网络中共有超级节点总数为 50。在 100W 节点规模的网络中, 超级节点个数为 500 个。设 10 万到 100 万规模的 P2PCSE 网络中超级节点数目为 D 。P2PCSE 系统限制转发过程中随机转发数目, 同时限定每个超级节点同时最多接受 3000 并发请求, 超出限制返回服务不可用标志, 直接拒绝请求。每个自治簇同时刻平均并发请求次数为 C , C 的范围为 $[0, 2000]$, 退出请求总数为 Q , Q 的范围为 $[0, 2000]$, 即网络中同时并发总数最大值为 $2000 + D * C$ 次。设搜索请求命令总数为 Y , 某超级节点接受到其他

D 个超级节点的转发命令请求, Y 与节点规模的关系为 $Y=D*C+2000$ 。经试验测试统计节点性能综合压力与并发规模如图 6-11 所示:

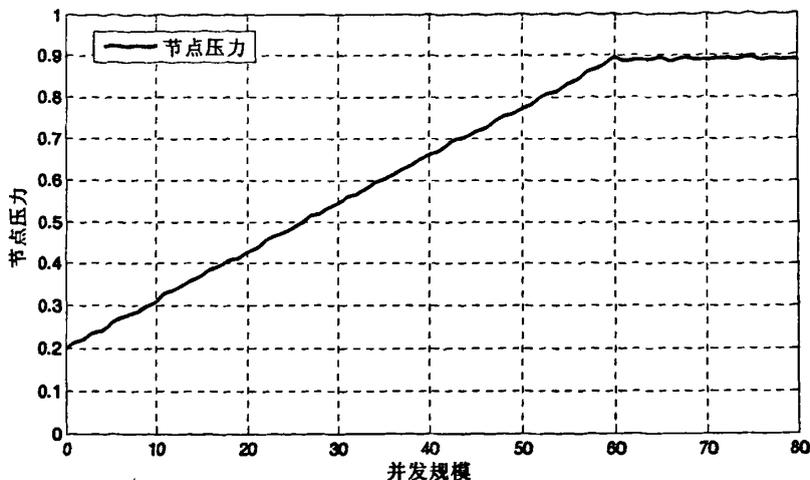


图 6-11 并发搜索时节点压力综合测试

其中 Y 轴代表节点的性能综合压力, X 轴代表并发请求规模, 单位为 50。从图 6-11 可以看出 P2PCSE 系统的搜索开销与节点并发规模(3000 内)是以相同趋势增长, 随着并发搜索请求(包括其他超级节点转发搜索请求)的增加, 超级节点所受压力增大。当来自普通节点及其他超级节点转发的并发请求达到最大值的时候, 超级节点压力达到最大值。因为系统限定超级节点最大接受其并发搜索请求的最大值, 随着并发请求的继续扩大, 节点压力基本同系统设定的最大值时节点压力相等, 从图上可以看出 P2PCSE 系统中超级节点的压力不随网络规模的扩大而扩大, 达到或超过系统设定安全并发值时, 节点压力始终保持稳定。

从图 6-11 上可以看出, P2PCSE 系统网络结构方案相比传统半分布式系统随着并发请求的增加, 超级节点压力并不一直随之增加以致节点崩溃, P2PCSE 系统具有很高的可靠性和稳定性。

6.5 本章小结

本章开篇详细分析了 P2PCSE 系统文件搜索流程, 介绍了 P2PCSE 系统各种情况下的界面。对 P2PCSE 全文检索子系统的关键部分——中文分词算法进行了对比测试, 对比采用 Chord 模型的软件进行搜索的准确度和满意度测试, 结果表明 P2PCSE 系统搜索准度和满意度比较突出。最后设定了模拟测试环境测试系统网络在各种并发请求下的综合性能,

得出 P2PCSE 系统的在网络具备稳定、可靠、可扩展的优点。

第七章 论文总结与展望

7.1 全文总结

本文针对中文搜索引擎技术及目前常见 P2P 搜索软件的网络结构进行国内外调研，提出了提出了一种基于 X-树结构的中文分词算法及改进的三层混合式半分布网络结构并在两者结合的基础上实现了 P2PCSE 原型系统。文章主要完成了如下几个方面的具体工作。

- 1) 详细分析了中文搜索引擎技术，并对影响中文搜索引擎搜索结果的关键技术——中文分词技术做了详细的解析和研究，指出了传统分词方法的不足。
- 2) 对比 P2P 模式与传统 C/S 模式的优缺点，归纳总结了常见 P2P 搜索技术采用网络结构模型，通过分析指出其缺陷。
- 3) 给出 P2PCSE 系统的总体设计，详细划分其模块并进行功能定义。对主要的通讯协议进行分析及实现。
- 4) 提出一种基于 X-树结构的中文分词算法 CSECHSplitor，对其进行详细的分析和算法实现，结合分词处理模块、文件处理模块、文件索引储存查询模块实现了全文检索子系统。
- 5) 提出一种改进的三层混合式半分布网络结构，详细分析了整体模型，给出节点定义及层与层之间的网络连接方式。并对构建该网络模型的关键部分进行了详细的介绍和实现。
- 6) 实现了 P2PCSE 原型系统，分析了系统文件搜索流程，对各种情况下的系统界面进行介绍。对 CSECHSplitor 算法性能，全文检索子系统的准确度与满意度，及网络结构在各种并发请求的性能下分别进行了测试，得出 P2PCSE 系统具备资源覆盖广、搜索结果准、网络可靠、稳定、可扩展的优点。

7.2 本系统的不足和未来工作展望

在实际的测试中，P2PCSE 系统具备资源覆盖广、搜索结果准、网络可靠、稳定、可扩展的优点。但分词处理模块中，英文和中英混合数据的处理存在不足，网络中超级节点的不确定性也是 P2PCSE 网络性能的另一缺点，同时在系统中超级节点限定最大并发请求为 3000，如何扩大这个限定值，更好的发挥 P2PCSE 系统的优点，在后继的工作中，我将继续这个方向的研究。

致谢

在本文即将完稿之际，我无法忘记那些直接或间接为论文做出贡献以及曾经给予我支持和帮助的人们。

首先，我要感谢的是我的家人，尤其是生我、养我并支持我学业的父母。在整整二十年的学业道路上，家人们给予了我最无私和最慷慨的最大支持。他们无微不至的关心和殷殷期盼，一直是我成长路上前行的动力。

特别地我由衷地感谢我的导师孙知信教授。导师渊博的专业学识，孜孜不倦、一丝不苟的治学精神以及亲切随和的待人接物，言传身教，给了我很大的教益。在我硕士三年的学习和毕业设计的全过程中，我始终得到了老师悉心的指导。老师深厚的理论水平、严谨的治学态度、朴素的生活作风及积极进取的精神给了我很大的教益，对我今后的人生道路产生了深远的影响。

我还要感谢诸位朝夕相处的项目组的小伙伴们，感谢他们在我学习、生活中给予的无私关怀和帮助。在整个项目的开发过程中，整个小组同甘共苦，一起讨论攻克了很多技术难关，在此向他们表示衷心的感谢。

最后，向各位不辞辛苦审阅本论文的教授、专家们表示由衷的谢意！

参考文献

- [1] 张军华,韩全会. 中文五大综合搜索引擎主要性能测评[J].情报科学,2008,9:1-4.
- [2] 方启明,杨广文,武永卫,郑纬民.基于 P2P 的 Web 搜索技术[J].软件学报, 2008, 19(10): 2706-2719.
- [3] Reynolds P, Vahdat A. Efficient peer-to-peer keyword searching[C]. In: Endler M, Schmidt D, eds. Middleware 2003. LNCS 2672, Berlin: Springer-Verlag, 2003. 21-40.
- [4] 刘琼,徐鹏,杨海涛等.Peer-to-Peer 文件共享系统的测量研究[J].软件学报, 2006, 17(10):2131-2140.
- [5] Zhou Jin,Li Kai, Tang Li,Towards a fully distributed P2P Web search engine[J],Proceedings - 10th IEEE International Workshop on Future Trends of Distributed Computing Systems, Proceedings - 10th IEEE International Workshop on Future Trends of Distributed Computing Systems, 2004:332-338.
- [6] Fang Qi-Ming , Yang Guang-Wen; Wu Yong-Wei et al,P2P web search technology[J],Journal of Software, 2008,19(10):2706-2719 .
- [7] Xiangji Huang, Aijun An,Design and implementation of a Chinese full-text retrieval system based on a probabilistic model[C],TENCON '93. Proceedings. Computer, Communication, Control and Power Engineering.1993 IEEE Region 10 Conference on,1993.
- [8] 曹勇刚,曹羽中,金茂忠 等.面向信息检索的自适应中文分词系统[J].软件学报,2006, 17(3):356-363.
- [9] 王亮. 搜索引擎及其相关性排序研究[D]. 武汉: 武汉大学, 2004.
- [10] 李晓明, 自宏飞, 王继明, 搜索引擎原理、技术与系统[M].北京:科学出版社 2005.30-50.
- [11] Page L,Brin S,Motwani R,Winograd T.The PageRank citation ranking:Bringing order to the Web.Technical Report[R],SIDL-WP-1999-0120,Stanford University,1999.
- [12] 张巍. 基于 PageRank 算法的搜索引擎优化策略研究[D].成都: 四川大学, 2005.
- [13] Wu K-J,Chen M-C,Sun Y. Automatic topics discovery from hyperlinked documents[J].Information Processing and Management,2004,40:239-255.

- [14] Al-Nuaimi M.O., Stephens R.B.L. Estimation of the effects of hilltop, singly distributed, trees on the path loss of microwave signals[J], Electronics Letters, 1997, 33: 873-874.
- [15] 陆一鸣, 胡健, 马范援. 一种基于源网页质量的锚文本相似度计算方法——LAAT[J]. 情报学报. 2005, 24(5): 548-554.
- [16] 马玉春, 宋涛瀚. web 中中文文本分词技术研究[J]. 计算机应用, 2004, 24(4): 134-136.
- [17] 易丽萍, 叶水生, 吴喜兰. 一种改进的汉语分词算法[J]. 计算机与现代化, 2007, 2: 13-15.
- [18] 梁刚. 基于机械分词与统计学的新词识别研究[J]. 情报理论与实践, 2005, 28(5): 475-477.
- [19] 邓宏涛. 中文自动分词系统的设计模型[J]. 计算机与数字工程, 2005, 33(4): 138-140.
- [20] 李玉梅, 陈晓, 姜自霞等. 分词规范亟需补充的三方面内容[J]. 中文信息学报, 2007, 21(5): 3-7.
- [21] Kai-Ying Liu, Jia-Heng Zheng. Research of automatic Chinese word segmentation[C]. Machine Learning and Cybernetics, Proceedings. 2002 International Conference on, 2002.
- [22] 吴栋, 滕育平. 中文信息检索引擎中的分词与检索技术[J]. 计算机应用, 2004, 24(7): 128-131.
- [23] 王蕾, 杨季文. 汉语未登录词识别现状及一种新识别方法介绍[J]. 计算机应用与软件, 2007, 24(8): 213-215.
- [24] 黄昌宁, 赵海. 中文分词十年回顾[J]. 中文信息学报, 2007, 21(3): 8-19.
- [25] 李知兵, 李龙澍. 基于数据驱动的中文分词方法研究[J]. 现代计算机, 2007(12): 8-10.
- [26] 邓祖明. P2P 搜索技术研究[D]. 合肥: 电子科技大学, 2007.
- [27] Subramanian R, Brian D G. Peer-to-Peer Computing: The Evolution of a Disruptive Technology[M]. London: Idea Group P, 2004: 61-63.
- [29] B. Pourebrahimi, K. Bertels, S. Vassiliadis. A Survey of Peer-to-Peer Networks [J]. Communications Surveys & Tutorials, IEEE. 2008, 3(2): 42-29.
- [30] Fox G. Peer to Peer Networks [J]. IEEE Computer in Science & Engineering, 2001, 3(3): 75-77.
- [31] 欧中洪, 宋美娜, 战晓苏等. 移动对等网络关键技术[J]. 软件学报, 2008, 19(2): 404-418.
- [32] Liu XM, Xiao L. hiREP, Hierarchical reputation management for peer-to-peer systems[C]. In: Proc. of the 2006 Int'l Conf. on Parallel Processing (ICPP 2006). 2006.
- [33] 何盈捷, 王珊, 杜小勇. 纯 Peer to Peer 环境下有效的 Top-k 查询[J]. 软件学报, 2005, 16(4): 540-552.

- [34] 夏启志,谢高岗.无结构 P2P 网络搜索方法及其改进[J].计算机应用研究,2005,22(9):256-260.
- [35] 唐辉,张国杰,黄建华,李祖鹏.一种混合 P2P 网络模型研究与设计[J].计算机应用,2005,25(3):521-524,535.
- [36] 陈志琦,苏德富.基于 P2P 技术的 Gnutella 网络搜索路由机制的改进[J].计算机工程与设计,2005,26(2):365-368.
- [37] 王涛,卢显良,侯孟书.Peer-to-Peer 网络中文件查询的研究[J].计算机科学,2005,32(10):49-51,145.
- [38] Ken Y.K. Hui,John C.S Lui,David K. Y.Yau.Small-world overlay P2P networks:Construction,management and handling of dynamic flash crowds[J].Computer Networks.2006, 50(15):56-59.
- [39] 杨峰,李风霞,余宏亮等.一种基于分布式哈希表的混合对等发现算法[J].软件学报,2007,18(3):714-721.
- [40] Joung Y, Fang C, Yang L. Keyword Search in DHT-based P2P Network [C]. Proc. of IEEE ICSCS'05. 2005: 1-10.
- [41] Meng Yong, Wang Xianbing. A DHT-based grid resource indexing and discovery scheme [C]. MITAlliance Symposium, Singapore. 2005.
- [42] Burstein P., Smith, A.J.;Efficient search in file-sharing networks[J].Parallel and Distributed Systems, 2007 International Conference on,2007,2:1 - 9.
- [43] 张谢华 李士峰.非结构化对等网络 Gnutella 搜索机制的改进[J].微计算机信息,2007,23(27):138-139,122.
- [44] 黄道颖,张安琳,等.分布式 Peer—to—Peer 网络 Gnutella 模型研究[J].计算机工程与应用,2003,39(5):60-63.
- [45] Xhafa F, Barolli L, Fernandez R et al,Extending JXTA Protocols for P2P File Sharing Systems[J].Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008. International Conference on, 2008 4-7:35 - 40.
- [46] Dougherty M, Kimm, H, Ho-sang Ham, Implementation of the Distributed Hash Tables on Peer-to-peer Networks[C], Sarnoff Symposium: 2008 IEEE ,2008.
- [47] 张龙,李巍,李云春.基于改进 DHT 算法的分布式资源发现模型的研究[J].计算机应用研究,2007,24(12):313-316.
- [48] 刘杰,王向辉,张国印.结构化 P2P 网络综述[J].信息技术,2007,31(6):143-144.

- [49] 陈冬芳,薛继伟,王征.基于对等模型的分布式协同设计系统[J].计算机工程,2007,33(15):43-45.
- [50] Zhao B.Y.,Ling Huang Stribling, J.et al.Tapestry: a resilient global-scale overlay for service deployment[J],Selected Areas in Communications, IEEE Journal on,Univ. of California, Berkeley, CA, USA,2004,22:41-53.
- [51] Dream tech software group. Peer to Peer Application Development: Cracking the Code [M]. Wiley Publishing. 2001.
- [52] 汪锦岭,金蓓弘,李京.结构化 P2P 网络上可靠的基于内容路由协议.软件学报,2006,17(5):1107-1114.
- [53] Stoica, I.; Morris, R.; Liben-Nowell, D. et al.Chord: a scalable peer-to-peer lookup protocol for Internet applications[J],Networking, IEEE/ACM Transactions on,2003 11:17-32.
- [54] 唐辉,李祖鹏,张国杰,黄建华.结构化覆盖网络模型 Chord 研究[J].计算机工程与应用,2005,41(1):132-135.
- [55] Dafan Dong, Ying Wu, Xuefei Wang et al, Similarity Search in Metric Space over Content Addressable Network[C].High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on,2008.
- [56] 海量公司,<http://www.hylanda.com/product/fenci/>[Z],天津,海量信息技术有限公司,1999.
- [57] Frans Kaashoek, David Karger.<http://pdos.csail.mit.edu/chord/>[Z],MIT,2008.