WWW分布数据源研究——数据模型和查询语言*

陈 澄 徐宏炳 王能斌

(东南大学计算机科学与工程系 南京 210096)

E-mail: ychen@seu.edu.cn

摘要 该文提出了分布式WWW(world-wide web)数据源WWWDS(WWW data sources)概念,详细介绍了WWWDS的数据模型.该数据模型包括节点、节点容器、节点图和链接点等,简单且具有扩展性,同时提出相应的查询能力强的查询语言WWWQL(WWW query language),并进一步探讨了查询处理和优化问题.

关键词 WWW,分布数据源,数据模型,查询处理和优化,查询质量.

中图法分类号 TP393,TP311

WWW(world-wide web)的出现迅速推动了Internet的发展,成为网络资源访问的标准模型,围绕WWW的新技术、新标准不断涌现,如Java语言、DBMS Web技术.前者为Web页面增加了动态特性,后者将传统的DBMS应用转移到以WWW为平台的新型Internet应用中,成为企业建立Intranet的基础.WWW创始人Tim Berners-Lee认为,WWW作为信息基础结构平台,不但存放大量企业数据,也会成为个人信息系统(Personal Information System)的平台[1],这些都促使WWW中的数据量呈爆炸性增长(据统计每年以10倍速度增长),从而引发了在庞大的WWW网络中数据的导航(Navigate)和检索问题.目前,解决这个问题主要有两种方法:① 通过创建可视导航图,图中标志各WWW节点及其链接关系,但只适于小型网络,不能反映Internet全貌;② 利用WWW资源发现技术和工具[2],如Alta Vista,Infoseek,Lycos等,它们利用一种称为Robot的资源自动发现进程,不断搜索相关WWW网页(通过深度优先或广度优先算法)来更新、维护索引数据库,为用户提供全文检索、约束性检索、基于布尔关系的查询方式,并对查询结果根据某种规则或算法评分(Rank)并排序.资源发现技术利用传统信息检索的成果很好地解决了如全文检索、索引维护等关键技术,是WWW的主要查询工具.但由于其出发点是作为网络用户工具,资源发现技术仍有以下不足:(1) 无查询语言,仅提供交互式表格驱动的查询,描述能力弱.(2) 没有充分利用WWW的主要语义信息——超文本链接.(3) 仅提供交互界面,未提供API,难以和其他系统集成,如关系数据库管理系统RDBMS和分布对象系统.WWW中含有丰富的信息,从浩瀚如海的数据中提取用户所需信息需要语法简单而描述能力强的查询语言,因此,需要在理论上为WWW建立统一的数据模型和查询语言.该数据模型应具有以下特点:(1) 模型简单,尽量自然表达WWW系统特点.(2) 准确刻画WWW的语义.(3) 支持功能强大的查询语言.(4) 可扩充性,适应WWW的发展.

美国国家自然科学基金会(NSF)在"21世纪数据库系统未来研究"研讨会(1995年5月)报告[3]中提出,WWW作为一个大型、自治的分布式系统,是数据库研究的新方向.本文详细讨论了WWWDS(WWW data sources)的数据模型和查询语言WWWQL(WWW query language)及其处理和优化.WWWDS概念及体系结构将另文探讨.

1 预备知识

1.1 WWW基础

WWW[4]是一个超媒体(Hypermedia)的全球信息检索系统,基于客户/服务器工作模型.客户和服务器之间通过超文本传输协议HTTP(hypertext transfer protocol)[5]交互信息,服务器称HTTPd(HTTP daemon),客户软件称为浏览器(Browser).客户通过全局资源定位器URL(universal resource locator)[6],向HTTPd请求取得文档,该文档具有不同类型,通过多用途Internet邮件扩展MIME(multipurpose internet mail extentions)[7]标准来标识.MIME用类型

---

* 本文研究得到国家自然科学基金资助.作者陈澄,1973年生,博士生,主要研究领域为数据库,网络.徐宏炳,1947年生,副教授,主要研究领域为计算机应用,数据库应用.王能斌,1929年生,教授,博士导师,主要研究领域为数据库,信息系统.

本文通讯联系人:陈澄,南京 210096,东南大学计算机科学与工程系

本文1997-06-02收到原稿,1997-08-05收到修改稿

/子类型(Type/Subtype)语法表示文档类型,如 image/gif 表示 GIF 格式图形文件. audio/wav 表示 WAV 音频文件. 其中最重要的是超文本标记语言 HTML(hypertext markup language)[8]文档,MIME 表示为 text/html.

例 1:一个简单的 HTML 文件.

```
<HTML>
<TITLE> World Wide Web </TITLE>
<BODY>
<H1><IMG SRC="images/WWWLogo.gif">Welcome to WWW</H1>
<A HREF="http://www.w3.org">(B) WWW Home</B></A>
</BODY>
</HTML>
```

其中标记<H1>...</H1>表示标题,<IMG...>是将图形嵌入显示,<B>...</B>表示字体加粗,<A HREF...>...</A>表示该处链接至 URL http://www.w3.org 所在的 HTML 文件,其中标记 A 表示锚(Anchor),它是 WWW 中超文本链接的关键. HTML 详细标准见文献[8].

1.2 WWWDS 简述

WWW 是由许多位于 HTTPd 服务器上的 HTML 文档通过链接、嵌入所组成的巨大的资源网络,可以视为一个全球性的分布数据源系统(Global Distributed Data Sources System over WWW),其中,每个 HTTPd 是一个高度自治的数据源,进而构成"联邦"数据源,本文称为 WWWDS. 它有以下特点:(1) 各数据源高度自治,基本无协调机制,形成"松耦合"联邦,联邦系统对各自治数据源无约束,联邦可大可小,伸缩性强,自治系统自由加入或退出联邦;(2) 数据源属于"同质"(Homogeneous),无异构性(Heterogeneous);(3) 数据源数据量大,数据种类多,生存期不一,变化大,进而引起链接失效或不一致(Inconsistent);(4) 数据中语义信息贫乏,链接是主要的语义信息,但可以认为(大多数情况下)无约束性的语言标记载有语义信息;(5) 各数据源更新、维护由各管理员操作,"联邦"用户仅能执行只读性的查询操作;(6) 联邦系统的数据量大,其查询操作不是"精确"型,即所有满足条件的节点都被检出,这对于 WWW 数据是不切实际的,因此,联邦系统有查询质量控制(QoQ),采用一种"尽力而为"的工作方式,在用户提出的质量要求下,返回贴近"精确"的结果.

2 WWWDS 数据模型

2.1 基本概念和定义

定义 1. 节点是二元组(ID,P),ID 唯一标志一个节点,P 是属性的集合,即 $P=\{$属性 $p_j, j=1,2,\ldots\}$.

节点代表了 WWW 中可被访问的一个具体文件,HTML 文件可以链接其他类型文件,称为主动型节点;其他只能被嵌入 HTML 的文件称为被动型节点. 属性集合 P 描述了该节点各种属性. 用<节点 ID>.<属性名>表示(以下 ID 用 N 或 $A,B,C,\ldots$ 表示). 主要属性有以下几种(括号为表示属性名).

(1) 相对 URL(url),表示该节点在相对 HTTPd 上的 URL 值. 如节点 http://www.seu.edu.cn/node.html 的相对 URL 为 node.html.

(2) 类型(type),节点类型,用 MIME 格式表示.

(3) 链接点集(link),是链接点(定义 3)的集合,仅主动型节点有此属性.

(4) 容器(container),是包含该节点的节点容器(定义 2)的 ID.

(5) 内容(content),泛指该节点所包含的内容,如例 1 的文件 N 中,"Welcome to WWW"$\in N$.content.

(6) 标记-权重表(tag-weight),是由元组(tag,weight)组成的集合. 其中 tag 表示 HTML 的标记,weight 为该标记的权重,取整数.

(7) 修改日期(update),表示节点最新的更新日期.

属性集合具有动态性,体现在:(1) 随节点不同类型而改变;(2) 随 WWW 发展而改变,如添加新属性.

定义 2. 节点容器是二元组(ID,P),ID 唯一标志一个节点容器,P 是属性集合. 节点容器代表了 WWW 中的一个 HTTPd,以下 ID 用 NC 表示. 主要属性有:

(1) 绝对 URL(url),表示该 HTTPd 的 URL 值.

(2) 节点集合(NodeSet),由该 HTTPd 中所有节点的 ID 组成的集合.

定义 3. 链接点是二元组$(ID,TargetID)$,ID 唯一标志一个链接点,$TargetID$ 是链接目标节点的 ID.

链接点仅存在于主动型节点内,其 ID 用 $\alpha,\beta,\chi,\ldots$ 表示,<链接点 ID>.$TargetID$ 表示链接目标节点(本文讨论

的粒度只到节点,实际上,HTML 中的锚可链接到节点内部). 若 $\alpha \in N.link$,且 $N=\alpha.TargetID$,称 α 为自环链接点. 若 $\alpha.TargetID$ 不存在,称 α 为断链接点.

定义 4. 链接 L 是三元组 (N_S,α,N_t),$\alpha \in N_S.link$,$\alpha.TargetID=N_t$.

若 α 是自环链接点,即 $N_S=N_t$,则链接 L 称为自环链接. 若 $N_S,N_t \in NC$,称 L 是容器内链接,否则,称为容器间链接.

定义 5. 节点图 NG 是一个二元组 (NV,NE),其中 NV 是节点 N 的集合,NE 是链接 L 组成的集合.

NG 构成多边有向图,即两节点间有多条相同的有向边,如图 1 所示.

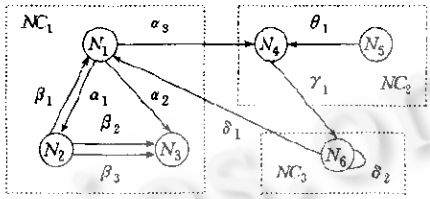

图1

其中 $NV=\{N_1,N_2,N_3,N_4,N_5,N_6\}$,$NE=\{(N_1,\alpha_1,N_2),(N_1,\alpha_2,N_3),(N_1,\alpha_2,N_4),(N_2,\beta_1,N_1),(N_2,\beta_2,N_3),(N_2,\beta_3,N_3),(N_4,\gamma_1,N_6),(N_5,\theta_1,N_4)(N_6,\delta_1,N_1),(N_6,\delta_2,N_6)\}$,$N_1,N_2,N_3 \in NC_1.NodeSet$,$N_4,N_5 \in NC_2.NodeSet$,$N_6 \in NC_3.NodeSet$. (N_6,δ_2,N_6)是自环链接.

定义 6. N 是孤立节点,若满足下面条件:不存在链接点 α,$\alpha.TargetID=N$.

如图 1,N_5 是孤立节点.

定义 7. 链接路径 LP 用链接的集合$\{(N_1,\alpha_1,N_2),(N_2,\alpha_2,N_3),\ldots,(N_m,\alpha_m,N_{m+1})\}$表示,$m>0$,其中 m 是路径长度. 特别地,当 $N_1=N_{m+1}$时,称 LP 为链接圈,用 LC 表示.

如图 1 所示,$(N_1,\alpha_1,N_2,\beta_2,N_3)$,$(N_2,\beta_1,N_1,\alpha_2,N_4)$是链接路径. $(N_1,\alpha_2,N_4,\gamma_1,N_6,\delta_1,N_1)$是链接圈.

若存在路径 $LP_1=\{L_{11},L_{12},\ldots,L_{1n}\} \subset$链接路径 $LP=\{L_1,L_2,\ldots,L_m\}$,称 LP_1 是 LP 的子路径,无子圈的链接圈称为平凡链接圈,以下若不特别声明,链接圈皆指平凡链接圈.

定义 8. 链接路径 $LP=\{(N_1,\alpha_1,N_2),(N_2,\alpha_2,N_3),\ldots,(N_m,\alpha_m,N_{m+1})\}$的节点集合 $NS_{LP}=\bigcup_{i=1}^{m+1}\{N_i\}$.

定义 9. 节点 N_1 对 $N_2(N_1 \neq N_2)$的链接耦合度 $LCD(N_1 \rightarrow N_2)=\|\{\alpha \mid \alpha \in N_1.link,\alpha.TargetID=N_2\}\|$,$N_1,N_2$ 的链接耦合度 $LCD(N_1,N_2)=LCD(N_1 \rightarrow N_2)+LCD(N_2 \rightarrow N_1)$.

若两节点存在链接,则它们必然存在着某种联系,链接耦合度在一定程度上反映了它们的联系强度. 如图 1 所示,$LCD(N_2 \rightarrow N_3)=2$,$LCD(N_3 \rightarrow N_2)=1$,$LCD(N_2,N_3)=3$.

定义 10. 节点容器 NC_1 对 $NC_2(NC_1 \neq NC_2)$的链接耦合度 $LCD(NC_1 \rightarrow NC_2)=\sum_{N \in NC_1}\sum_{N' \in NC_2}LCD(N \rightarrow N')$,$NC_1$,$NC_2$ 的链接耦合度 $LCD(NC_1,NC_2)=LCD(NC_1 \rightarrow NC_2)+LCD(NC_2 \rightarrow NC_1)$.

定义 11. 有路径 LP,若存在链接圈 $LC_1,LC_2,\ldots,LC_n$,$n>1$,使 $LP \subset LC_i$,且不存在 $LC' \neq LC_i$,使 $LP \subset LC'$,则节点集合NS_{LP}的链接聚合度 $LAD(NS_{LP})=n$. 若$(N_1,N_2,\ldots,N_n)$不能构成链接路径,则 $LAD(\{N_1,N_2,\ldots,N_n\})=0$.

链接聚合度反映了多个节点的关联程度,n 值越大,说明这些节点间关系越大. 显然,若 $LP_1 \subset LP_2$,则 $LAD(NS_{LP_1}) \geqslant LAD(NS_{LP_2})$.

2.2 基本操作

WWWDS 模型中主要对象是节点.因此操作都是对节点实施的.

(1) 选择操作 σ.

σ 的一般形式是 σ(节点集合 NS,选择条件 $Cond$),其结果是节点集合 NS_{result}.

例 2:$\sigma(\{A,B,C,D\},\text{“}Database\text{”} \in content \wedge x \in link \wedge \text{“}Object\text{”} \in x.content)$,表示在节点 A,B,C,D 中选择满足以下条件的节点:节点中有字符串 $Database$,且有至 x 节点链接,x 有字符串 $Object$. 其中选择条件 $Cond$ 中出现的 x 称为形式节点,$x \in NS$. 条件中不带限定部分的属性(如 $content$,$link$)是无约束属性,指选择操作当前正在处理节点的属性.

例3:选择与节点N位于同一节点容器的节点集合:$ID \in (N.container).NodeSet$.

不难证明,等式$\sigma(\ldots(\sigma(\sigma(NS,Cond_n),Cond_{n-1})\ldots),Cond_1)=\sigma(NS,Cond_1 \wedge Cond_2 \wedge \ldots \wedge Cond_n)$成立.

(2) 节点投影操作Π

投影操作Π是节点集合NS到节点容器集合NCS的映射.定义为$\Pi(NS)=\{NC|$存在$N \in NS, N.Container=NC\}$,$NCS$称为$NS$的投影.

(3) 容器扩张操作Ψ

扩张操作Ψ是节点容器集合NCS到节点集合NS的映射.定义为$\Psi(NCS)=\bigcup_{NC \in NCS} NC.NodeSet$,$NS$称为$NCS$的扩张.

显然有$NS \subset \Psi(\Pi(NS))$,$NCS=\Pi(\Psi(NCS))$.

(4) 集合操作

指节点集合的一般集合操作,如交、并、差等.

2.3 约　束

WWWDS模型中约束如下.

(1) 节点ID、节点容器ID、链接点ID都是唯一的,它们唯一标志相应的节点、节点容器和链接点.

(2) 一个节点不能属于多个节点容器,一个链接点不能属于多个节点.

(3) 节点属性集合与节点类型相关.如非主动型节点无链接集合属性,仅有HTML文件有标记-权重表属性.

3　查询语言WWWQL

WWWQL建立在WWWDS数据模型基础之上,每条查询语句都可以化为相应的数据模型操作.使用WWWQL令用户不必在错综复杂的WWW网中"航行"搜索,而是基于内容的搜索(Content-based Search).WWWQL是一种类SQL的说明性语言,包括查询语句和辅助定义语句.

3.1 查询语句

查询语句语法如下.

SELECT NODEURL|CONTAINERURL

[INTO〈节点集合标识符〉|〈节点容器集合标识符〉]

[FROM All | {〈节点〉|〈节点容器〉|〈节点集合标识符〉|〈节点容器集合标识符〉|〈别名〉[,〈节点〉|,〈节点容器〉|,〈节点集合标识符〉|,〈节点容器集合标识符〉|,〈别名〉]* }]

[PREFER {〈节点〉(〈整数〉)|〈节点容器〉(〈整数〉)|〈节点集合标识符〉(〈整数〉)|〈节点容器集合标识符〉(〈整数〉)|〈别名〉(〈整数〉) [,〈节点〉(〈整数〉)|,〈节点容器〉(〈整数〉)|,〈节点集合标识符〉(〈整数〉)|,〈节点容器集合标识符〉(〈整数〉) |,〈别名〉(〈整数〉)]* }]

WHERE〈条件表达式〉

其中NODEURL表示选择结果为节点的绝对URL值,CONTAINERURL表示节点容器的绝对URL值.选择结果集合可以分别用节点集合标识符或节点容器集合标识符表示.FROM子句中是查询语句的查询范围,是节点的集合,称为待选节点集合(CNS),用相应的URL表示.All表示系统中所有节点,节点容器表示节点容器的扩张,集合标识符表示相应的节点或节点容器集合.FROM子句缺省时等价于FROM All.PREFER子句中(整数)表示该节点是重要节点,称为查询节点权重,该值越大表示节点越重要.WHERE后表示条件表达式,其正规语法限于篇幅,不详细给出.下面通过举例说明.

例4:选择所有含有字"Computer"和"Language",且本月更新过的HTML节点.

SELECT NODEURL WHERE Computer,Language IN Content(10) AND

type=TEXT/HTML AND month(update)=month(today())

其中IN相当于$\in$;(10)表示Computer,Language在节点中出现的权值.因为HTML文件中不同标记中的字重要程度可能不同,一般情况下,加粗的字**Computer**比未加粗的Computer更重要;在题头出现的字比正文中更重要等等.不同标记的权值可以通过辅助定义语句定义.type=TEXT/HTML,表示仅对HTML类节点查询,缺省时表示对所有类型节点.函数month()返回日期的月份,today()表示今天的日期.

例5:选择所有节点url中包含Database的节点容器.

SELECT CONTAINERURL WHERE Database IN url

例6:选择所有满足下面条件的HTML节点N:含有"Object Oriented",且有链接至含有"OODB"的HTML节点

N',N''包含链接至 N. 应着重注意节点 http://www.omg.org.

SELECT NODEURL WHERE "Object Oriented" IN Content AND type=TEXT/HTML AND
x IN *link* AND *ID* IN x.*link* PREFER http://www.omg.org(10)

该查询语句中引入了形式节点 x.

条件表达式的包含操作符 IN 中可以包含另一个查询,即嵌套查询,但针对 WWW 查询的逐步求精特点,可以用流查询功能替代.

定义 11. 一个流查询 SQ 是查询 $Q_1,Q_2,\ldots,Q_n$ 组成的序列,记为 $SQ=\langle Q_1,Q_2,\ldots,Q_n\rangle$. 其中 Q_i 可以使用 $Q_1,Q_2,\ldots,Q_{i-1}$的结果.

流查询优点在于当前查询范围可以利用前期查询的结果(中间结果),特别适用于交互式查询,可以让查询者逐渐缩小查询范围,最终得出精确结果.

例 7:流查询举例.

```
BEGIN
//(1) 选择含有字 COMPUTER 和 LANGUAGE 的节点容器集合 LANGUAGE
      SELECT CONTAINERURL INTO LANGUAGE WHERE "COMPUTER","LANGUAGE" IN Content
//(2) 选择含有字 INTERNET 的节点容器集合 INTERNET
          SELECT CONTAINERURL INTO INTERNET WHERE "INTERNET" IN Content
//(3) 从 COMPUTER 扩张和 LANGUAGE 扩张的交集中选择含有 JAVA 的节点集合
        SELECT NODEURL WHERE ID LANGUAGE AND ID INTERNET AND"JAVA"IN Content
END
```

3.2 辅助定义语句

3.2.1 别名定义语句

节点和节点容器的标识 *ID* 唯一标识相应实体,其值一般与 URL 有关(以下举例时假设 *ID* 即 URL). 因为 URL 较长,在书写查询语句时不方便. 因此,引进别名定义语句. 其语法如下.

DEFINE 〈节点〉[,〈节点〉]* |〈节点容器〉[,〈节点容器〉]* AS〈别名〉

例 8:DEFINE http://WWW.W3.ORG AS WWWHOME.

例 9:DEFINE http://WWW.OMG.ORG/corba.html,http://WWW.IONA.COM/corba.html AS CORBA.

别名定义后,在查询语句中别名等价于其所定义的节点或节点容器的集合.

3.2.2 标记-权重定义语句

该语句用于定义 HTML 语言中不同标记中字的权重. 语法如下.

DEFINE TAG 〈标记〉 |default WEIGHT 〈权重值〉[ON 〈节点名〉|〈节点容器名〉]

其中标记部分 default 表示无标记情况,权重值 $w\geqslant 0$,$w=0$ 表示该标记中的字可忽略. ON 子句表示定义在特定节点或节点容器上,缺省表示查询时的所有节点.

例 10:DEFINE TAG 〈B〉 WEIGHT 4 //粗体标记

例 11:DEFINE TAG 〈I〉 WEIGHT 2 //斜体标记

例 12:DEFINE TAG default WEIGHT 0 //无标记

利用带权重的查询可以提高查准率.

4 查询处理和优化

4.1 查询处理

查询处理首先将查询语言转化为相应的节点操作,即选择、投影、扩张、交、并和差等,将 FROM 子句中的节点容器扩张,并放入待选节点集合 *CNS*,然后依次扫描待选节点集合 *CNS*,将满足条件的节点 N 放入结果集合中,同时,对结果做评分(Rank).

4.1.1 单查询处理

查询处理中用到的数据结构说明如下.

(1) 待选节点表(CNL). 存放待选集合 *CNS* 的列表.

(2) 结果表(RL). 存放结果节点或节点容器的列表.

(3) 标识符-节点集合表(INT). 存放二元组(标识符,节点集合),用于存放别名和节点集合对应关系.

设查询条件 $Conditon=Cond_1$ OR $Cond_2$ OR...OR $Cond_n$,其中 $Cond_i$ 中仅含原子项(形如 $a=b$,NOT $a=b$ 的项)

的逻辑与(AND)操作.其中 $Cond_k$ 共有 $n(k)$ 个原子项,不失一般性,设前 $m(k)$ 个不含有形式参数,项过程 SELECT(*CNS*,*Condition*)处理查询操作,返回结果集合,其算法如下.

(1) $RL=\varnothing$,$k=1$;

(2) 根据 *CNS* 初始化 *CNL*,按查询节点权重排序,$temp=\varnothing$;

(3) 若 *CNL* 非空,从 *CNL* 取下一节点 *N* 为当前节点;否则转(8);

(4) 判断当前节点是否满足 $Cond_k$ 的前 $m(k)$ 项,若不满足,转(3);

(5) $m'=n(k)-m(k)$,若 $m'=0$,转(7);

(6) 从 m' 个含有形式参数的项中选一形式参数 x.设 $Condition'=Cond_{m+1}$ OR $Cond_{m+2}$ OR...OR $Cond_n(k)$,将 *Condition'* 中无约束属性替换为当前节点 *N* 的相应属性值,将形式参数 x 的属性替换为无约束属性,设替换后条件为 *Condition'*(x),递归调用 $temp'=$SELECT(*CNS*,*Condition'*(x)).若 $temp'=\varnothing$,转(3),

(7) $temp=temp\cup\{N\}$,转(3);

(8) $RL=RL\cup temp$,$k=k+1$;若 $k<n+1$,转(2);

(9) 若所求为节点容器集合,返回 $\Pi(RL)$;否则,返回 *RL*.

4.1.2　流查询处理

对流查询,用到以下数据结构:查询链表 QL 是由查询 $Q_1,Q_2,\ldots,Q_n$ 组成的链表.过程 *S_SELECT*(*QL*)处理流查询,返回结果集合,算法如下.

(1) $RL=\varnothing$;

(2) 若 QL 非空,从 QL 中取下一查询 *Q* 为当前查询;否则,转(5);

(3) *RL*=SELECT(*Q* 的 *CNS*,*Q* 的 *Condition*);

(4) 若 *Q* 查询有 INTO 子句,将节点集合标识符和 *RL* 插入 INT 表中,若 *RL* 是节点容器集合,将节点容器标识符和 $\Psi(RL)$ 插入 INT 表中;转(2);

(5) 若所求为节点容器集合,返回 $\Pi(RL)$;否则,返回 *RL*.

4.1.3　评分处理

评分(rank)是用一个量化的数值(分数)衡量结果集合中节点的相对重要程度.例如,满足条件 computer∈content(10)的节点集合 RNS 中,computer 的权重虽然都大于 10,但权重高的节点可能更"重要"一些.本文提出以下几种评分算法思路(设对于某一查询 *Q*,节点 *N* 的得分为 *rank*(*Q*,*N*),*rank*(*Q*,*N*)=0 表示该节点不满足查询条件):

(1) 利用传统信息检索(IR)中的技术.IR 领域中需要解决被检索文档与查询的"相似度"(Similarity),这与评分概念类似.比较典型的方法是 TFxIDF 方法,它将查询和文档分别表示成向量空间模型(Vector Space Model)中的查询向量和文档向量,利用距离公式计算相似度.

(2) 基于查询节点权重的评分.查询节点权重是用户指定的"重要"节点,也是用户关心的节点.这些节点应获得较高的分数.

(3) 基于标记权重的评分.权重大的得分高,这特别适用于针对内容的查询.各标记权重可以用权重定义语句定义,因此,在不同定义下的评分标准不同,可以根据需要调整.

(4) 基于链接耦合度的评分.当节点 *N* 得到较高分数时,与节点 *N* 耦合度较大的 *N'* 的分数亦上升(*N'* 也必须是结果集合中节点).

(5) 基于链接聚合度的评分.当节点 *N* 得到较高分数时,与节点 *N* 处于同一较高聚合度路径上的节点分数亦上升(*N'* 也必须是结果集合中节点).

(6) 基于形式节点个数的评分.满足节点 *N* 条件的形式节点数目越多,*rank*(*Q*,*N*)越大,对含有形式节点条件的查询作用较大.

(7) 对节点容器的评分.节点容器 *NC* 的分数与满足条件的容器中节点的分数有关.设 $N_1,N_2,\ldots,N_m\in NC.NodeSet$,且 $rank(Q,N_i)>0$,则 $rank(Q,NC)=f_1(m)$ 或 $rank(Q,NC)=f_2(\max(rank(Q,N_1),rank(Q,N_2),\ldots,rank(Q,N_m)))$.

4.2　查询优化

WWWDS 中查询和一般数据库查询要求不同,最大的特点是,WWWDS 的查询不要求精确结果,而是在查询质量(QoQ)控制下的"尽力而为"型的查询,因为在庞大的待选节点集合 *CNS* 中选择出所有节点是不切实际的.查询优化的目标就是在效率和精确两者中取得最佳效果,可以分为静态优化和动态优化,本文仅讨论静态优化.在查询质量控制(QoQ)下的动态优化技术将另文介绍.

4.2.1　基于条件表达式的优化

基于条件表达式的优化主要是在判断条件表达式时尽量缩小 *CNS*,提前约束形式节点.以下讨论时,设 *Condition*

是由原子项的与构成,其前 m 项没有形式节点,后 m' 项含有形式节点.

(1) 逻辑表达式化简,提取公因项.如"*computer*"∈*Content* ∧ *title*="*computer*" ∨ *title*="*computer*"等价于 *title*="*computer*".

例 13:"*database*"∈*Content* ∧ *title*="*computer*" ∨ *title*="*computer*" ∧ "*language*"∈*Content*,等价于 *title*="*computer*" ∧ ("*database*"∈*Content* ∨ "*language*"∈*Content*),这样,可以先求条件 *title*="*computer*",使待选节点数目减少.

(2) 先判断不含属性 *Content* 的原子项,因为对内容搜索相对其他属性比较慢,若涉及非 *Content* 属性的原子项条件不满足,则不需判断含 *Content* 属性的原子项.

例 14:"*computer*","*language*"∈*Content* ∧ *title*="*computer*"可以优化为 *title*="*computer*" ∧ "*computer*","*language*"∈*Content*.

(3) 提前约束形式节点.形式节点的数目直接影响查询效率.可以事先约束可以约束的形式节点,较大程度地缩小形式节点的待选范围.

例 15:"*computer*"∈*Content* ∧ $x.$∈*link* ∧ $x.title$="*language*",可以先用原子项 $x.title$="*language*"约束形式节点 x 至小范围.

4.2.2 简化形式节点的选择操作

在 4.1 节的 SELECT 算法中可以看出,形式节点集合是否为空决定着当前节点是否满足要求,因此,并不需要求出满足要求的所有形式节点,所以,可以改进算法.当前节点属于形式节点时,求出一个满足节点即可返回.

4.2.3 基于节点容器的优化

当查询结果要求为节点容器集合时,表明用户只想找到节点容器 ID,并非该节点容器中的所有节点,因此,在查询处理时,若当前节点投影已属于 RL,则不需处理,这种方法可以大大减少需扫描的节点数.

4.2.4 基于链接耦合度和链接聚合度的优化

查询往往受查询质量(QoQ)控制约束,如查询时间、结果个数等,因此,必须在约束下得到较优的解.链接耦合度 LCD 在一定程度上反映了两节点的联系强度,可以根据与当前节点链接耦合度从大到小的顺序处理形式节点,这样,满足条件的形式节点可以尽早出现,加快查询速度.在选出一个节点时,继续检查与该节点链接耦合度大或与该节点同处于高链接聚合度路径的节点.

WWWDS 中的查询处理和优化有一定的难度,有时甚至可能存在矛盾,如基于形式节点个数的评分和简化形式节点的优化,关键在于如何在 QoQ 约束下,对不同类型查询灵活处理,利用静态和动态优化结合缩小 CNS,提高查询效率.

5 总 结

WWWDS 数据模型基于节点、节点容器、链接点和节点图,准确地刻画了分布式的 WWW 数据源的特点,并且具有可扩展性,如对新类型节点的加入或属性的增加等,并提出节点链接耦合度、链接聚合度等定义,在理论上定性描述了节点间的关系.本文在数据模型基础上提出了类 SQL 的查询语言 WWWQL,它提供了比目前 WWW 资源发现系统更强的查询能力,还提供了针对 WWWDS 的流查询功能.模型的操作还可以进一步扩充,支持更强能力的查询.本文中节点集合选择操作结果仍是节点集合,可以扩展为 n 元组节点集合(节点集合可以视为一元组集合,是特例),这样就可以支持较复杂的查询"找出所有节点对(x,y),其中 x 节点含有 *computer*,y 节点含有 *database*,且 x 与 y 之间互相链接",这种查询方式已经超过了传统 WWW 资源的发现范围,对支持在 WWW 环境下的数据挖掘有一定意义,但这就引起了操作上的不封闭,因此,在描述上有一定难度,查询处理和优化也必然不同.

本文提出的数据模型、查询语言、查询处理和静态优化算法为构建分布式 WWW 数据源系统奠定了理论基础,使 WWWDS 可以方便地和其他系统集成.如目前 Galaxy[9]系统正在利用对象技术实现异构数据源集成,各数据源通过包装器(Wrapper)采用 OMG CORBA[10]标准实现互操作,WWWDS 数据模型可以较方便地转化为对象模型,为系统提供公共的对象层接口.WWWDS 系统结构及其它相关技术将在另文中进一步探讨.

参考文献

1 Tim Berners-Lee. WWW: past, present, and future. IEEE Computer, 1996,29(10):69~77

2 Stacey Kimmel. Robot generated DATABASES on the World Wide Web. DATABASE, Feb. 1996,19(1):41~49

3 Avi Silberschatz, Michal Stonebraker, Jeffery D Ullman. Database research: achievements and opportunities into the 21st

century. SIGMOD RECORD, Mar. 1996,25(1):52～63

4 Tim Berners-Lee, Cailliau R, Loutonen A *et al*. The world-wide web. Communications of ACM, 1994,37(8):76～82

5 Tim Berners-Lee, Fielding R, Frystyk H. Hypertext Transfer Protocol HTTP/1.0, Internet Draft(1995). http://www.w3.org/pub/WWW/Protocols/HTTP1.0/draft-ietf-http-spec.html

6 Tim Berners-Lee. Uniform Resource Locator. 1992. http://www.w3.org/hypertext/WWW/Addressing/Addressing.html

7 MIME (Multipurpose Internet Mail Extensions). Internet Engineering Task Force (1993) RFC1521. ftp://ds.internic.net/rfc/rfc1866.txt

8 Tim Berners-Lee, Connolly D W. Hypertext Markup Language 2.0. Internet Network Working Group RFC1866, MIT/W3C (1995). ftp://ds.internic.net/rfc/rfc1866.txt

9 王宁,陈滢,俞本权等.Galaxy 异构数据源集成系统的设计.见:唐常杰编,数据库进展 97'——全国第 10 界数据库会议论文集.成都,1997
(Wang Ning, Chen Ying, Yu Ben-quan *et al*. Galaxy, a system for integrating heterogeneous data sources. In, Tang Chang-jie ed. Proceedings of the Data Base Conference'97. Chengdu, 1997)

10 Object Management Group. The Common Object Request Broker: Architecture and Specification, Revision 2.0. July 1995

Towards Globalization of Distributed Data Sources over WWW
——Data Model and Query Language

CHEN Ying　XU Hong-bing　WANG Neng-bin

(*Department of Computer Science and Engineering　Southeast University　Nanjing*　210096)

Abstract　In this paper, the concept of global distributed data sources over WWW (WWWDS) is introduced. A simple and extensible data model based on node, node container, node graph and link points are discussed in details. A powerful SQL-like query language (WWWQL) for WWWDS is introduced and the query processing and optimization technology for WWWQL are discussed as well.

Key words　World-Wide web, distributed data sources, data model, query processing and optimization, quality of query.