

面向大型网络的多管理对象信息的组预取算法*

张金祥¹, 吴建平¹, 韩淑玲¹, 吴泉源², 王怀民²

¹(清华大学 信息网络工程研究中心,北京 100084);

²(国防科学技术大学 计算机学院,湖南 长沙 410073)

E-mail: jxzhang@cernet.edu.cn

http://www.tsinghua.edu.cn

摘要: 如何减少网络管理系统对网络带宽的占用是网络管理研究的难点之一.结合管理信息库(management information base,简称 MIB)信息的收集策略直接影响着网管系统性能的特点,针对异构大型网络环境下多管理对象的信息收集问题,提出了一种从多个被管设备中提取有关 MIB 信息的新算法——组预取算法(group-prefetching algorithm,简称 GPA).该算法能在不改变现有的网络管理协议的情况下,将多个分散的、在未来一段时间内的 MIB 对象的访问组合在一个预取组中进行访问,减少对管理对象的访问频度,从而可以较少地消耗所用的网络带宽和 CPU 资源,减小整个网络开销.

关键词: 网络管理;MIB;信息提取;组预取

中图法分类号: TP393 **文献标识码:** A

目前,计算机网络发展的规模性和异构性使得网络管理日益重要^[1].强大的网管功能可以保证网络正常的运行.但同时,网管功能又需要消耗大量的网络资源,从而增加了网络开销.如何减少网管系统对网络带宽的占用是网络管理研究的难点之一.在网管系统中,管理功能都是管理进程与代理间通过交换管理信息库 MIB(management information base)中定义的管理对象信息来实现的,因此,访问管理信息的方式直接影响着网管系统的性能^[2].

在异构、多被管对象网络环境下,针对多个不同被管设备上的被管对象,通常采用顺序固定、时间间隔固定、离散、单个对象访问算法(sequence-and interval-fixed discrete single access algorithm,简称 SIFD)来提取管理信息,其缺点是,既存在多个小对象同时被离散访问的情况,又存在访问对象的空轮询现象.这样,当被管对象较多时,会占用大量网络资源(耗时、需大量传输带宽),性能低^[3],并且可能造成网络拥塞.例如,中国国家邮政计算机网在 1999 年 4 月的时候,网管流量曾一度占了有效带宽的 30%左右,迫使人地关闭网管系统^[4].

针对以上缺点,IETF 在 SNMP v2.0 中增加了支持提取大量数据的 Get Bulk 操作^[5].但是该方法采用非原子操作,不能保证传输的正确性,缺少有效的对象选择组合策略和容错策略,并且其效果与 Get Next 一致,基本只局限于访问表类型的 MIB 对象信息^[6],因此在实际中并未得到广泛应用.

本文提出的组预取算法 GPA,基于现有的管理协议,如简单网络管理协议 (simple network management protocol,简称 SNMP)、公共管理信息协议 (common management information protocol,简称 CMIP)、互操作协议 (Internet inter-operability protocol,简称 IIOP),针对所有类型 MIB 对象的访问,将若干个在未来一段时间内离散

* 收稿日期: 2000-12-02; 修改日期: 2001-06-05

基金项目: 国家自然科学基金资助项目(69833030, 69725003, 90104002);国家高技术研究发展计划资助项目(863-306-ZD02-01-2, 2001AA112041)

作者简介: 张金祥(1969 -),男,甘肃泾川人,博士,讲师,主要研究领域为分布计算,网络管理,网络安全;吴建平(1953 -),男,山东巨野人,博士,教授,博士生导师,主要研究领域为网络协议测试;韩淑玲(1971 -),女,甘肃甘谷人,硕士生,主要研究领域为软件工程,网络安全;吴泉源(1942 -),男,上海人,教授,博士生导师,主要研究领域为分布计算,智能软件;王怀民(1962 -),男,江苏南京人,博士,教授,博士生导师,主要研究领域为分布计算,智能软件.

存取的小对象作为一个整体访问.同时,采用容错策略、组成员动态变化策略及自适应策略,根据网络当前的通信量,可自适应地调整组中对象及轮询时间,避免空轮询.既保证了管理信息传输的正确性,又能显著降低管理对象的访问频度,因而使轮询次数较少,减少了网管系统对网络资源的占用.

1 组预取算法 GPA

1.1 问题描述

GPA 算法涉及到的问题是:设异构网络中有 n 个被管对象,属于 m 个 Agent 管理($n \gg m$),则算法根据本次对 n 个被管对象轮询的结果值、被管对象的属性状态和时间特性,从中选取 r ($r < n$) 个被管对象作为一组,并组合成一个大的访问报文,一组对象作为一个整体被更新,而且每组只要一个类标识符,以标识该组对象的时间特性.同时,动态地调整组中对象的个数及轮询的下次间隔,以合理的轮询频度取得所有的期望值.其中,属性状态表示被管对象所属的 Agent 进程的运行状态,其值由算法与事件进程通过消息通信来维护;时间特性是指被管对象的值更新对时间的敏感程度.

1.2 GPA 的概念及功能

GPA 算法是指将 MIB 对象按照与时间的相关性分类,对时间特性一致的若干个小的对象的报文访问,组成一个或多个大的报文,当用户请求该组中的某个对象值时,把整个大报文都发出,从而可以返回整组对象的值,并把结果保存在本地 Cache 中,此后用户请求时就可以直接从 Cache 中取对象的信息而不必通过网络访问获得.同时,根据本次访问 MIB 对象所得出的网络通信量和延时,自适应地调整组中 MIB 对象和读取 MIB 数据的策略,以期较少地消耗所用的网络带宽和 CPU 资源.

GPA 是为了在广域网环境中以组预取的方式轮询多台被管设备有关被管对象的信息.这些信息可用商品数据库进行存储,从而为网络系统的监控、流量计费和性能分析等提供可靠依据.GPA 算法应具备的基本功能有: 算法应具备容错功能,保证管理信息的正确传输; 算法实现的程序应具有自动判断被管对象在网络中所处的物理位置的功能,只将物理位置靠近且时间特性一致的对象作为一个访问组; 算法实现的程序应具有可伸缩性,可以动态增加或减少组所含的对象; 算法尽可能少地占用网络管理工作站(network management station,简称 NMS)的处理资源,不明显降低 NMS 处理数据和请求的能力.

1.3 GPA 算法中的几个关键策略

1.3.1 组预取策略

根据网络管理对被管对象的时间要求及被管对象值的特点,将被管对象分为 3 类^[2]:

(1) 实时性数据 R_Data(real-time_data).要求及时地轮询被管对象的值,以期获得反映被管对象最近特性的数据,这类对象一般不采用 Cache 策略.如用于计费管理的表示网络中通信量记录的 actPkts 被管对象的特性值数据信息需要及时采集;

(2) 非实时性数据 N_Data(not-real-time_data).对被管对象的值轮询一次后,此值在该对象的使用期限(time to live,简称 TTL)内均有效.在 TTL 内,收到对该被管对象的访问请求时,返回 Cache 中的数据;否则,从实际的目标被管对象实例存取值.如表示路由表的被管对象的值不必频繁地采集;

(3) 时间无关性数据 I_Data(irrespective-time_data).这种被管对象的值一经采集,就存于 Cache 中,一般不必再更新,如表示日志入口和系统描述信息的被管对象的值为静态信息.

根据上述对被管对象的分类,在开发网络管理应用系统的过程中定义被管对象时,每一个被管对象需要增加一个说明其时间特性的属性,为了减小由此而带来的被管对象的额外存储开销,也便于系统的扩展和编程,我们对时间特性进行二进制编码,用一个字节中的两位表示所定义的时间特性 Tattr(time attribute),即

0(00):R_Data, 1(01):N_Data, 2(10):I_Data.

而把该字节中的其余 6 位作为保留位,留作它用(如用于组的标识).

R_Data 类型的数据由于对时间敏感性很强,一般采用收到请求就立即离散访问的方式.而 N_Data 和 I_Data

类型的数据都适合于采用组预取策略获取信息,记 $S(\text{cur_g})$ 为当前访问的组对象对应的集合,其时间特性为 $S(\text{cur_g})^{\text{Tattr}}$, any_o 为该组中任一对象, any_o 的使用期限为 $\text{any_o}^{\text{TTL}}$,令该组的未来时间段(即组的使用期限)为 $S(\text{cur_g})^{\text{TTL}}$,则:

$$S(\text{cur_g})^{\text{TTL}} = \min\{\text{any_o}^{\text{TTL}} \mid \forall \text{any_o} \in S(\text{cur_g})\}; \text{如果 } S(\text{cur_g})^{\text{Tattr}} = 1, \\ S(\text{cur_g})^{\text{TTL}} = \text{任意值}; \text{否则,如果 } S(\text{cur_g})^{\text{Tattr}} = 2,$$

即 N_Data 型对象组成的组预取的使用期限值选为该组中对象的 TTL 的最小者,而 I_Data 型对象由于其与时间基本无关,因而组成的组的使用期限值可选为任意值.

对于 N_Data 和 I_Data 型对象,每一组需要一个属性用以指定其时标 $Tstamp(\text{time stamp})$,即该组上次被更新时的系统时间.

设当前系统时间为 cur_systime ,记当前访问的组对象对应的集合 $S(\text{cur_g})$ 的时标为 $S(\text{cur_g})^{\text{Tstamp}}$,若接收到用户对该组中任一被管对象的请求时,则根据以下策略预取该组中所有被管对象的值:

从被管系统中取值并更新 Cache 中值和时标;

$$\text{如果 } (\text{cur_systime} - S(\text{cur_g})^{\text{Tstamp}}) \geq S(\text{cur_g})^{\text{TTL}}$$

从 Cache 中取值;否则

即如果当前系统时间与组时标之差大于等于该组的使用期限 $S(\text{cur_g})^{\text{TTL}}$,则收到对组中某一被管对象的访问请求时,从实际的目标被管对象访问组中所有对象的值,同时对组中被管对象的 Cache 中的值进行更新,并相应地修正该组的时标;否则,返回 Cache 中被请求的被管对象的数据.

1.3.2 组对象自适应扩充策略

根据本次访问 MIB 对象所得出的网络通信量和延时,可自适应地调整预取组中 MIB 对象的数目,从而以有效的对象选择策略实现被管对象信息的组预取.

记某一被管网络中所有的被管对象组成的集合为 $S(\text{all_g})$,所观察的对象为 any_o , $\text{any_o} \in S(\text{all_g})$,且 any_o 不属于任何预取组,记本次访问 any_o 的往返时间 (round trip time,简称 RTT) 为 $\text{any_o}^{\text{RTT}}$,时间特性为 $\text{any_o}^{\text{Tattr}}$.令当前访问的组对象对应的集合 $S(\text{cur_g})$ 中对象数为 $S(\text{cur_g})^{\text{Count}}$,则自适应扩充策略要求:

若 $\text{any_o} \notin S(\text{cur_g})$ 且 $S(\text{cur_g})^{\text{Count}} < \text{Max_gsize}$,则

$$S(\text{cur_g}) = S(\text{cur_g}) \cup \{\text{any_o}\}, S(\text{cur_g})^{\text{TTL}} = \min\{S(\text{cur_g})^{\text{TTL}}, \text{any_o}^{\text{TTL}}\};$$

$$\text{如果 } \text{any_o}^{\text{RTT}} < \text{Alarm_Time} \text{ 且 } \text{any_o}^{\text{Tattr}} = S(\text{cur_g})^{\text{Tattr}}$$

$$S(\text{cur_g}); \text{否则}$$

即以本次访问对象的 RTT 与预先设定的时间阈值 Alarm_Time 相比较,若 RTT 远远小于 Alarm_Time 且与组的时间特性一致,则把该对象加入到组中.其中, Max_gsize 为预先设定的组对象个数的上限,以避免组无限增大而产生网络拥塞.

同时,对当前访问的组对象对应的集合 $S(\text{cur_g})$ 组的使用期限 $S(\text{cur_g})^{\text{TTL}}$ 进行修改,即取原组的使用期限与新加入组的被管对象的 TTL 的较小值为新组的使用期限.

1.3.3 组对象自适应收缩策略

若 $\text{any_o} \in S(\text{cur_g})$,则

$$S(\text{cur_g}) = S(\text{cur_g}) - \{\text{any_o}\}; \text{如果 } \text{any_o}^{\text{RTT}} \geq \text{Alarm_Time}$$

$$S(\text{cur_g}); \text{否则}$$

即以本次访问对象的 RTT 与预先设定的时间阈值 Alarm_Time 相比较,若 RTT 大于等于 Alarm_Time ,则从组中把该对象删除,而后使用离散定时读取 MIB 数据的方法;当 RTT 降低后,再恢复使用组预取策略.同时,对组的使用期限进行相应的调整.

1.3.4 容错策略

一组访问对象中有可能有某些对象所处的被管设备或者系统因关闭等原因而无法响应,从而导致返回的信息不完全.这就要求算法对这些错误进行分析,对于因被管系统造成的错误,则从该组中把所对应的对象删除;对于因协议造成的数据包丢失,则重新采集,同时将出错信息和处理动作均写入告警日志文件 Error.log ,以

恢复出对象的真实信息.

1.3.5 动态识别对象策略

设置两个配置表,记录当前处于活动状态设备的表 act_t 和记录当前组对象对应的设备信息表 cur_t.算法只对属于 act_t 中的 Agent 所管理的组对象进行组预取访问,保证不会对休眠的被管设备进行轮询.即

若 $any_o \in S(cur_g)$, 则

$S(cur_g) = S(cur_g) - \{any_o\}$; 如果 $any_o^{id} \notin act_t$

$S(cur_g)$; 否则

其中 any_o^{id} 表示 any_o 所属的 Agent 的标识 id 号.实现时,我们用 Agent 的 IP 地址来代表其 id 号.当某被管设备活跃时,可通过事件系统将该对象动态加入到 act_t 中.同时可自动检测对象状态,即算法对某被管对象查询时,若该对象数次无响应,则将此对象从预先设定的 act_t 表中删除.从而可以减小网络资源的浪费.

原则上,在不考虑网络突发性拥塞的情况下,物理位置靠近、大小一致的请求报文其 RTT 应该基本相同.因此,本文采用 RTT 及对象的 id 值来判断对象在物理位置上是否靠近,若 id 相同,则对象处于同一被管设备上;否则,若 RTT 相同或相近,且对象所属 Agent 属于同一管理域,则可视为物理位置靠近.

1.4 GPA算法描述及实现

在采用 GPA 算法实现广域网环境下的被管对象的信息组预取时,我们依照功能和被管设备所处位置的不同,分为多个管理域.在每个管理域处设置相应的子管理者 Sub_Manager^[7].若为离散访问报文,则 Manager 直接向相应的目的 Agent 发请求并接收返回的结果;若为组预取报文,则 Manager 向相应的 Sub_Manager 发大请求报文,再由 Sub_Manager 从各 Agent 处获取管理信息后,把结果以链接的方式返回给 Manager.

GPA 所用的数据结构定义如下:

```
typedef struct_ObjStruct {
    char        *name;
    int         type;
    int         oid_length;
    ObjectID    oid[MAX_SUBID_LEN];
    unsigned char Tattr;        /*时间特性*/
    int         RTT;           /*访问本对象的往返时间*/
    int         Group_label;   /*所属组标号,初值为 0,表示不属于任何组*/
} ObjStruct;    /*定义 MIB 被管对象的数据结构*/

typedef struct _InfoStruct {
    ObjStruct    *obj;
    bool         Status;      /*属性状态*/
} InfoStruct;    /*定义 S(cur_g)元素的数据结构*/
```

对一组被管对象进行组预取的 GPA 算法可用类语言描述如下:

1. begin
2. if $(cur_systime - S(cur_g)^{Tstamp}) < S(cur_g)^{TTL}$
3. 返回 Cache 中该组对象的相关值
4. else begin
5. 将 $S(cur_g)$ 组中对象组成循环队列
6. 取出 $S(cur_g)$ 组中所有对象 oid, 并组成一个大的请求报文
7. 取出相应 Sub_Manager/Agent 的 IP 地址, 并与该 Sub_Manager/Agent 建立一个 Session
8. if 三次均失败 then 错误信息写入 Error.log.Exit
9. 采用组预取策略获取该组中 MIB 对象的信息

10. 根据返回结果的时延修正各对象的 RTT 值
11. if 存取对象信息有错
12. then 采用容错策略以离散方式定时采集对象信息
13. 采用组对象自适应扩充和收缩策略动态改变组对象
14. 修正 act_t 表的内容,并按动态识别对象策略动态改变组对象
15. end
16. 将对象信息写入数据库
17. end

2 性能测试

在“中国移动通信网本地网网管系统”的详细设计和开发中,我们已经利用本算法的基本思想实现了异构网络中不同的被管网元设备中与性能管理、计费管理相关的被管对象管理信息的采集,并且在实验室网络环境下,对利用 Visual C++ 实现的 GPA 算法进行了性能测试。

测试环境为并行与分布处理国家重点实验室 PDL(National Parallel and Distributed Laboratory)内线路速率为 10Mb/s 的局域网 Intranet,所用的网管平台是在基于公共对象请求代理体系(common object request broker architecture,简称 CORBA)的分布计算软件平台 StarBus 上实现的多级网络管理系统 MHNMs^[2].测试对象为 $m=3$ 个被管结点,即 Intranet 的 Windows NT 服务器、Windows 98 平台以及 Windows 95 平台。

用 CORBA IDL 分别定义了 50 个被管对象用于描述局域网中被管理 Windows NT 服务器、Windows 98 和 Windows 95 平台的各个特性.针对同一组被管对象的定义,我们把采用 GPA 算法策略的管理与采用 SIFD 策略的管理进行了访问性能测试比较。

取 Alarm_Time=30ms,Max_gsize=40.测试中的报文数包括 Manager 与目录服务器交互以获取 Sub_Manager/Agent 的 IP 地址的报文和请求/应答报文,且 Manager 与 Sub_Manager/Agent 建立一个 Session 后,此后对 Sub_Manager/Agent 所属被管对象的访问均用此 Session。

2.1 同一被管结点

在对同一被管结点上管理信息的组访问测试时,取 50 个被管对象中的 Obj_Number 个作为一组进行实验,得到的一组测试数据见表 1。

Table 1 Packets in single network node
表 1 单个结点通过网络传输的报文

Obj_number	Packets number		Rate
	SIFD	GPA	
10	22	14	0.64
18	38	22	0.58
21	44	25	0.57
30	62	34	0.55
36	74	40	0.54
38	78	42	0.53

测试的被管对象数, 通过网络传输的报文数 顺序固定、时间间隔固定、离散的单个对象访问算法, 组预取算法, 采用 GPA 算法的网管报文数与采用 SIFD 算法的网管报文数的比率。

表 1 中的数据说明,随着组中对象个数的增加,在 Obj_Number 个对象属于同一引用物理位置时,采用 GPA 策略的网管报文数与一般离散的 SIFD 策略的网管报文数差距成倍增大。

如表 1 中的 Obj_Number 为 10,当即有 10 个被管对象满足属于同一预取组的条件时,离散的 SIFD 策略管理通过网络传输的报文数为 22 个,采用 GPA 策略管理的报文数为 14 个,两者之比 Rate 为 0.64. 采用 GPA 策略管理通过网络访问被管对象的报文个数远远少于 SIFD 策略的管理(约为 0.64 倍)。

2.2 多个被管结点

在对多个被管结点上管理信息的组访问测试时,分别从 3 个被管结点中选取若干个对象作为一组进行实验,得到的一组测试数据见表 2.

Table 2 Packets in multi network nodes
表 2 多个结点通过网络传输的报文

Obj_number	Packets number		Rate
	SIFD	GPA	
10	26	18	0.69
18	42	26	0.62
21	48	29	0.60
30	66	38	0.56
36	78	44	0.56
38	82	46	0.56

测试的被管对象数, 通过网络传输的报文数 顺序固定、时间间隔固定、离散的单个对象访问算法, 组预取算法, 采用 GPA 算法的网管报文数与采用 SIFD 算法的网管报文数的比率.

表 2 中的数据说明,随着组中对象个数的增加,在 Obj_Number 个对象分别属于不同引用物理位置时,采用 GPA 算法策略的网管报文数与一般离散的 SIFD 策略的网管报文数差距也成倍增大.

如表 2 中的 Obj_Number 为 10,我们分别从 3 个结点上选取 3 个、4 个和 3 个对象共 10 个合成一组,即有 10 个被管对象满足属于同一预取组的条件时,离散的 SIFD 策略管理通过网络传输的报文数为 26 个,采用 GPA 策略管理的报文数为 18 个,两者之比 Rate 为 0.69.采用 GPA 策略管理通过网络访问被管对象的报文个数远远少于 SIFD 策略的网管(约为 0.69 倍).

上述两类测试结果均表明:采用 GPA 算法的网管报文数比一般离散的 SIFD 算法的网管报文数显著减少.

3 结束语

本文提出的组预取算法 GPA,采用容错策略、组预取策略、有效的组对象自适应扩充和收缩策略以及动态识别对象策略,将对时间特性一致的多个 MIB 对象的离散访问组合成一个预取组.既保证了管理信息传输的正确性,又能显著降低管理对象的访问频度,减少了网管流量对网络带宽的占用.这对在多管理对象的广域网中以合理的方式、无丢失地采集多台被管设备有关 MIB 对象信息,从而实现综合网络管理功能具有重要的作用.

References:

- [1] Redlich, J-P., Masaaki, Suzuki, Weinstein, P. Distributed object technology for networking. IEEE Communications Magazine, 1998,36(10):100~111.
- [2] Zhang, Jin-xiang, Wu, Quan-yuan, Wang, Huai-min, *et al.* Design and implementation of a corba-based multi-hierarchy network management system. Journal of Computer Research and Development, 2001,38(7):793~797 (in Chinese).
- [3] Jayant, R.H., Michael, O.B., *et al.* MANDATE: managing networks using database technology. IEEE Journal on Selected Areas in Communications, 1993,11(9):1361~1372.
- [4] The Information Bureau of China Post. The overall design of application software for China post computer network. Technique Report, 1999 (in Chinese).
- [5] Case, J., McCloghrie, K., Rose, M., *et al.* Introduction to community-based SNMPv2. Technique Report, RFC 1901, 1996.
- [6] Rose, M., McCloghrie, K., Davin, J. Bulk table retrieval with the SNMP. Technique Report, RFC 1187, 1990.
- [7] Zhang, Jin-Xiang. The research and implementation of object-oriented distributed network management technology [Ph.D. Thesis]. Changsha: National University of Defence and Technology, 2001 (in Chinese).

附中文参考文献:

- [2] 张金祥,吴泉源,王怀民,等.一种基于 CORBA 的多级网络管理系统的设计与实现.计算机研究与发展,2001,38(7):793~797.

- [4] 国家邮政局信息技术局.全国邮政综合计算机网应用软件总体设计.技术报告,1999.
[7] 张金祥.面向对象的分布式网络管理技术的研究与实现[博士学位论文].长沙:国防科学技术大学,2001.

A Group-Prefetching Algorithm of Information for Multi-Managed Objects in WAN*

ZHANG Jin-xiang¹, WU Jian-ping¹, HAN Shu-ling¹, WU Quan-yuan², WANG Huai-min²

¹(Network Research Center, Tsinghua University, Beijing 100084, China);

²(School of Computer Science, National University of Defence Technology, Changsha 410073, China)

E-mail: jxzhang@cernet.edu.cn

<http://www.tsinghua.edu.cn>

Abstract: It is increasingly a difficult problem of how to decrease the usage of network resources by network management systems. Otherwise, the performance of network management is affected by the scheme to collect MIB (management information base) information. A new algorithm, GPA (group-prefetching algorithm), is put forward, which is used for retrieving MIB information from the managed devices in heterogeneous WAN (wide area network). GPA can adjust many little retrieving objects into one prefetching group to access, decrease the frequency of retrieval, and thus can use fewer network resources and decrease the network overhead without alteration of the existing network management protocols.

Key words: network management; MIB (management information base); information retrieval; group-prefetching

* Received December 2, 2000; accepted June 5, 2001

Supported by the National Natural Science Foundation of China under Grant Nos.69833030, 69725003, 90104002; the National High-Tech Research and Development Plan of China under Grant Nos.863-306-ZD02-01-2, 2001AA112041