维普资讯 http://www.cqvip.com

基于结构化 P2P 网络的计算力资源整合方法的研究与实现*)

钮 麟 刘 业 庄艳艳 刘林峰

(东南大学计算机网络和信息集成教育部重点实验室 南京 210096)

(东南大学计算机科学与工程学院 南京 210096)

摘 要 结构化 P2P 网络因其查找效率高、资源消耗少的特点受到了越来越多的关注,但由于缺少易于开发和应用的平台,结构化 P2P 网络上的应用还没有广泛出现,这限制了结构化 P2P 网络的普及。本文研究了结构化 P2P 网络在计算力资源整合应用上的优势,提出了一种基于结构化 P2P 网络的计算力资源整合方法,面向系统性能、可靠性、缩放性进行了优化,设计并开发了相应的原型系统。对该方法进行了理论分析和实验验证。结果表明,这一方法有效地整合了网络中节点的计算力,并对网络动态变化和负载均衡有较强的适应能力。

关键词 结构化 P2P 网络,计算力资源整合,负载均衡

Research and Realization of Computing Power Integration in Structured P2P Networks

NIU Lin LIU Ye ZHUANG Yan-yan LIU Lin-feng

(The key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing 210096, China)

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

Abstract Structured P2P networks have attracted more and more attention for their high efficiency and low resource consumption. Due to the lack of a platform for easy development, the applications of structured P2P networks have not been widely appeared which limits their popularization. This paper studies the advantages of integrating computing power based on structured P2P networks and proposes a novel methodology to integrate computing power in structured P2P networks, develops a prototype which is optimized for system performance, reliability and scalability. Analysis is done in both theoretical and experimental scenarios. The results show that this methodology integrates the computing power of the nodes in the network effectively, and is adaptive to the dynamics of the network and load balancing.

Keywords Structured P2P networks, Computing power integrating, Load balance

1 引言

近年来,P2P 网络作为一种新型网络资源整合形式,受到了广泛的关注。根据节点间拓扑结构的不同,P2P 网络可以分为非结构化 P2P 网络和结构化 P2P 网络。非结构化 P2P 网络的节点间的联系是随机产生的,其节点查找消息主要通过泛洪(blooding)或随机漫步(random walk)的方法在节点间传播,具有查找效率低、网络资源消耗大的特点,但因其实现简单,运行稳定而得以广泛的应用。这类 P2P 网络最经典的应用是文件共享,如 Gnutella。为了改进非结构化 P2P 网络的不足,研究者开始关注结构化 P2P 网络。结构化 P2P 网络节点间的联系通过一定的拓扑规则产生,因此节点与其邻节点具有特定的逻辑关系,这种关系因路由算法的不同而不同。由于节点间的逻辑关系,节点查找消息可以按照特定的路由路径到达目标节点,具有查找效率高、资源消耗少的特点。结构化 P2P 网络支持具有哈希表语义的 lookup 操作,因此也被称为分布式哈希表(Distributed Hash Table, DHT)。目前已经出现的较为典型的结构化 P2P 算法有 Chord[1], Pastry[2], Taperstry[3], CAN[4], Viceroy[5]等。结构化 P2P 网络因其优良的性能和较少的资源消耗在学术界受到了很大的重视。

但是,基于结构化 P2P 网络的应用服务的研究较为薄弱,目前仅有 Chord/DHash(http://pdos.csail.mit.edu/chord/)项目中所开发的网络存储系统原型。要进一步推动结构化 P2P 网络的发展,除了在可用性部署能力等方面继续改进结构化 P2P 算法以外,还需要加强利用结构化 P2P 网络提供各类应用服务的研究。大部分结构化 P2P 网络都具有一定的负载均衡能力,这使结构化 P2P 网络非常适合应用于需要负载均衡的环境。计算力资源整合指在单台计算机无法满足计算任务对计算力要求的情况下,整合网络中其它计算机的闲置计算力资源以共同完成计算任务。它是一种对负载均衡具有较高要求的服务,如著名的 BOINC(http://boinc.berkeley.edu/)系统就是这样一个计算力资源整合系统。本文认为,除了在负载均衡上的优势外,与其它计算力资源整合方法相比,利用结构化 P2P 网络进行计算力资源整合还具有以下的优势:(1)结构化 P2P 网络是完全自组织的,没有服务器的存在。网络中的成员在提供计算力的同时,也能利用网络的总体计算力为自己服务,节点间是完全平等的,这使 Internet 上的公共用户更愿意参与进来,从而获得更强的总体计算力。(2)接受服务的节点不需要了解网络中节点的数量、具体节点的状态等信息,逻辑上可以将整个网络视为一台具有强大计算能力的超级计算机,这有助于减少节点管理和索引的开销。

目前结构化 P2P 网络上的计算力资源整合的研究尚未见文献全面报道。本文将对结构化 P2P 网络上的计算力资源整合方法进行研究,并结合研究成果设计一个分布式的大质数测

*)本研究工作受到国家自然科学基金项目:面向服务的 Overlay 网络及其关键技术的研究(编号 60573133)的资助。

维普资讯 http://www.cqvip.com

试程序。本文的其余部分是这样组织的:第2节系统阐述基于结构化P2P网络的计算力资源整合方法,从理论和工程上对系统模型、计算任务的分解、分配、结果汇总,网络通信开销的控制,提高可靠性,适应动态变化的计算力和提高负载均衡能力方面进行深入的研究;第3节结合第2节的成果开发一个分布式大质数测试系统;第4节以该系统为基础,测试并分析本文所提出的方法的各项性能;最后总结本文的研究成果。

2 结构化P2P网络下的计算力资源整合方法的研究

2.1 系统模型

计算力资源的整合方法有很多。对于结构化P2P网络这样的松耦合系统,分治法是一种较好的思路。分治法将一个大型计算任务分解成一定数量的工作片断,分解后得到的工作片断按一定的规则分配给网络中的节点,节点根据当前的运算负荷决定是否接受计算任务,并向任务发出者发送计算结果或拒绝通知。最后,任务发出者综合各工作片断的计算结果,导出原计算任务的结果。在这个计算模型下,至少有下列问题需要解决:(1)如何将一个计算任务分解、分配到网络节点,并将工作片断的结果汇总成最终的计算结果;(2)就计算任务本身而言,网络通信属于一种"额外开销",如何使通信开销不成为系统的瓶颈;(3)P2P网络是一个动态环境,节点的构成相当复杂,采取何种保证措施才能使每一个工作片断都得到有效的处理(即得到计算并返回结果);(4)随着节点的动态加入与退出以及节点本身状态的变化,网络的总体计算力在不断变化,如何适应这种变化以使资源利用最大化;(5)在网络节点数量无法获取以及节点间具有很强异构性的情况下,如何分配任务使得网络中各节点的负载处于相对平衡的状态。

这些问题是基于结构化P2P网络的计算力资源整合的关键问题,本文将对这些问题进行深入的研究,并提出一系列解决方案。

2.2 计算任务的分解、分配与结果汇总

计算任务的分解与结果汇总很大程度上与具体的计算任务有关,大部分情况下属于数学问题,很难找到一个通用的方法,本文对此不做深入讨论。在第3节的原型系统中,对于具体的问题,本文设计了问题分解与结果汇总方法。

计算任务分解为工作片断后,紧接着要解决的问题是工作片断的分配,即将工作片断交由某一网络节点来处理。分配的主要原则是负载均衡。一般而言,结构化P2P网络本身就具有一定的负载均衡能力,如Chord算法使用了SHA-1散列来生成它的key,使得key分布较为均匀。本文采用的方法是,对工作片断的二进制表示做散列操作,得到与特定结构化P2P算法长度要求一致的key,并将工作片断分配给负责维护这一key的节点来完成计算。这种方法在结构化P2P网络中具有通用性,并且可以获得初步的负载均衡能力。本文后续部分进一步研究如何获得更高的系统性能和资源利用率。

2.3 控制网络通信的开销

处理一个工作片断的开销包括如下几个部分:网络通信(节点查找与任务分配)时间、排队时间、计算时间、结果返回与处理时间。结果返回与处理时间非常短,不属于本文研究的重点,因此工作片断总的开销实际上由前面三个部分组成。网络通信时间指结构化P2P网络查找负责与工作片断对应的key的节点并将工作片断发往该节点的时间。如果节点正在忙于处理其它工作片断,会将当前工作片断置于等待队列中,这就是排队时间。排队时间受节点当前负荷的影响,但可以用第2.6.2节提出的语义队列的思想加以控制。工作片断排队期间,节点实际上并未闲置,可以不认为是额外开销。因此,网络通信时间成为了唯一的额外开销。遗憾的是,网络通信时间是由P2P网络本身(包括网络状况、节点的数量、节点加入退出的频率等)决定的,上层应用很难从根本上减少,只能动态地适应网络。因此,目标可以转化为:合理控制网络通信时间占总开销的比重。在总开销一定的情况下,比重太低,则表明单个工作片断的计算时间长,节点负荷过重;比重太高,则表明大量的时间花费在了网络通信上,计算力资源利用率低。因此要选择一个合适的比重,既能降低节点负荷,又能保证较高的资源利用率。本文设计的控制网络通信时间所占比重的算法如下。

首先实时地获取总开销中各个部分的耗时。处理工作片断的节点记录工作片断的计算时间 $t1$ 和排队时间 $t2$,并在结果返回的报文中包括这两个值。工作片断发送端测量工作片断的即时周转时间 iTT(Instant Turnaround Time),并计算网络通信时间 $iCT=iTT-t1-t2$。根据每个工作片断的即时周转时间,可以实时地计算出系统当前的平均周转时间。设系统当前的平均周转时间为 TT,收到第 n 个工作片断的计算结果后,可测出相应工作片断的即时周转时间 iTT_n,根据如下公式更新 TT 值:

$$TT_n=TT_{n-1}\times k+iTT_n\times(1-k)\quad(0<k<1)$$

不妨标记收到的各个工作片断的周转时间分别为 iTT_0,iTT_1,…,iTT_n,令 $TT_0=iTT_0$,展开上式得:

$$\begin{aligned}TT_n&=TT_{n-1}\times k+iTT_n\times(1-k)\\&=(TT_{n-2}\times k+iTT_{n-1}\times(1-k))\times k+iTT_n\times(1-k)\\&=TT_{n-2}\times k^2+iTT_{n-1}\times(1-k)\times k+iTT_n\times(1-k)\\&=iTT_0\times k^n+(1-k)\times\sum_{j=1}^{n}iTT_j k^{n-j}\quad(n>0)\end{aligned}$$

从以上分析可以看出,当前的 TT 值受最近的 iTT 值影响较大,这样的计算方法使得计算得到的 TT 值更能反映网络的近况,有利于适应网络的动态变化。用同样的方法可计算平均网络通信时间 CT。获得了以上数据后,在系统运行过程中,可以实时监测到当前网络通信时间所占的比重 $T=CT/TT$,并设定该比重的上限 u、下限 v。当 $T>u$ 时,增加后继工作片断的计算量;当 $T<v$ 时,减小后继工作片断的计算量(计算量的增减方法与具体应用相关)。在实际实现中,为了进一步减少瞬间值带来的误差,只有当 $T>u(T<v)$ 的状态维持一个设定的时间 t 后,才调整后继工作片断的计算量。第4.1节通过实验验证了这一方法的可行性。

2.4 利用重传机制提高可靠性

由于P2P网络的复杂性,有时会出现节点查找失败、报文丢失、节点拒绝请求等情况,这些都将使发送者无法收到相应工作片断的计算结果,导致整个任务无法完成。为了适应这样的环境,本文提出了使用重传机制提高系统可靠性的方法。

需要重传的工作片断分为两类:一是被节点拒绝的工作片断,二是在指定时间内未收到计算结果的工作片断,即超时。两种情况下,任务发出者都采用二进制指数退避法执行重传。但在第二种情况下,需要确定初始的超时时间。

在第2.3节中,已经给出了实时获取当前系统平均周转时间 TT 的方法。初始超时时间设定为当前 TT 的 c 倍($c>1$)。对于上述算法中 k 和 c 的具体数值,可以不做规定,由具体的计算任务根据实际情况确定。重传机制的介入,确保了所有的

维普资讯 http://www.cqvip.com

工作片断都能得到计算并被响应，提高了系统的可靠性。

2.5 适应动态变化的计算力

结构化 P2P 网络的特性决定了任务发出者无法准确获知网络中节点的具体数量和总体计算力。在运行过程中，随着节点的加入和离开，以及各节点本身的状态变化，网络的总体计算力呈现动态变化的态势。在这种情况下，任务发出者无法确定连续向网络中发送多少工作片断较为合适，因此需要一种机制使工作片断的发送速率与网络的总体计算力动态匹配。否则，节点只能以固定的速率向网络发送工作片断。速率过慢，总体计算力得不到充分的发挥；速率过快，则造成网络中的节点负载过高，大量工作片断被拒绝或丢包率增高，不利于充分、合理地利用总体计算力。本文借鉴 TCP 协议中的慢启动和拥塞控制算法来解决上述问题[6]。在计算力资源整合中，发生拥塞的主要标志是任务发出者收到节点的拒绝通知，这通常是由于已分配但未得到处理的工作片断的总计算量超过了网络当前的总体计算力，任务发出者应降低发出工作片断的速率。本文针对这一特点设计了一个简化的发送速率控制方法。

任务发送者设置一个发送窗口，保证其已发送但未得到处理的工作片断数量不超过发送窗口的大小。初始发送窗口大小设置为 1，每收到一个计算结果，就增加窗口大小。提供计算力的节点设置一个等待队列，当队列中的工作片断数量到达队列长度时，拒绝新到的工作片断，以免其负担过重。当任务发出者收到拒绝通知时，表明网络中节点已经过载了，此时减小发送窗口的大小。待新的计算结果收到时，再逐渐增大窗口大小，以适应总体计算力的变化。窗口大小将维持在一个动态平衡状态，以拟合网络的总体计算力变化。发送窗口每次减小和增大的具体数值属于系统策略问题，不属于本文的研究重点。引入了发送速率控制机制后，任务发送者将随着网络总体计算力的变化自动地以适当的速率发送工作片断，使其能够充分合理地利用网络节点的总体计算力。第4.2和4.3节通过实验验证了这一算法的有效性。

2.6 提高负载均衡能力

结构化 P2P 算法中 key 的分布一般是均匀的。但相对于 key 的数量(如 Chord 算法为 2^{160})而言，网络中的节点数量较少，无法保证节点的分布是完全均匀的。在这种情况下，势必会有一部分节点负责较多的 key，另一部分节点负责较少的 key。如果某一节点所负责的 key 数量很多，而其所负责的 key 又被频繁请求，该节点就会过载，在网络中形成热点(HotSpot)[7]。热点的存在会严重影响网络的整体性能。在 P2P 网络中，网络节点的构成非常复杂，既有性能一般的普通 PC，也有性能高上几个数量级的高性能集群。根据节点当前的工作状态，其闲置计算力也处于动态的变化之中，这就是节点间的异构特性。这些因素决定了工作片断不应以绝对平均的方法分配给每一个节点，而要根据计算力的差异动态地进行分配。本节将提出解决上述两个问题的方案。第 4.4 节给出了改进前后负载均衡能力的对比。

2.6.1 工作片断重定向

节点过载的标志是等待队列已满而新的工作片断还在不断地到达。为了缓解节点的负荷，并增强负载均衡能力，本文引入重定向机制。具体工作过程如下：当节点的等待队列满而新的工作片断到达时，将该工作片断重定向到节点的某一邻居节点(如 Chord 中的后继节点)。若该邻居节点仍然过载，则再次将工作片断重定向到下一个邻居节点，直至未过载节点。引入重定向机制后，负责 key 较多的节点可以将工作片断转发到其它节点上去，减轻本节点的负荷；负责 key 较少的节点有机会接收从负责 key 较多的节点转发来的工作片断，增加本节点的负荷，从而在整个网络范围内改善节点和计算任务分布不均匀带来的负载不均衡问题。

如果整个网络都处在过载的状态中，工作片断将会在网络中循环，造成不必要的开销。在实际实现中，每一个工作片断的请求报文中都设置跳数限制字段，其初始值由任务发出者填入，工作片断每次被重定向时，该字段都会减小 1。当该字段值减小到 0 时，如果仍然需要重定向，则最后一个处理该工作片断的节点将向任务发出者发出拒绝通知。如第 2.5 节中所描述的，任务发出者收到拒绝通知后，将调整工作片断的发送速率，以降低整个网络的负荷，且被拒绝的工作片断将在一段时间后被重新分配。下面以 Chord 算法为例，从理论上分析利用重定向机制提高负载均衡能力的原理。假设网络中的节点总数为 M，总工作片断数量为 W。理想情况下，每个节点接收的工作片断数量 N 如下式所示：

$$N=W/M \tag{1}$$

实际上，由于节点分布不均，造成每个节点接收到的工作片断的数量不同。设节点 i 接收到的工作片断数量为 N_i。在不启用重定向机制的情况下，各节点收到的工作片断数量的标准差 S_1 如公式(2)所示：

$$S_1=\sqrt{\frac{1}{M}\sum_{i=0}^{M-1}(N_i-N)^2} \tag{2}$$

启用了重定向机制后，节点将工作片断重定向到其后继节点。假定重定向的最大跳数限制为 h，理想情况下，每个工作片断被转发 0 次，1 次，…，h 次的概率为 $1/(h+1)$。在实际的动态网络中，这个概率会动态地变化，并且不会在节点间平均分布。可以假定每个工作片断被发 0 次，1 次，…，h 次的概率分别为 $k_0/(h+1)$，$k_1/(h+1)$，…，$k_h/(h+1)$，且 $\sum_{i=0}^{h}k_i=h+1$。在这个模型下，节点 i 收到的实际工作片断总数可表达如下：

$$T_i=N_i\times\frac{k_0}{h+1}+N_{(i-1+M) \bmod M}\times\frac{k_1}{h+1}+\cdots+N_{(i-h+M) \bmod M}\times\frac{k_h}{h+1}=\sum_{j=0}^{h}N_{(i-j+M) \bmod M}\times\frac{k_j}{h+1} \tag{3}$$

各节点收到的工作片断数量的标准差为

$$S_2=\sqrt{\frac{1}{M}\sum_{i=0}^{M-1}\left(\sum_{j=0}^{h}N_{(i-j+M) \bmod M}\times\frac{k_j}{h+1}-N\right)^2} \tag{4}$$

标准差 S 越小，表明网络中各节点完成的工作片断的数量差别越小，即负载均衡能力越强。在式(2)中，没有任何可以人为控制的变量。在式(4)中，当 h 趋向于正无穷大时，每个工作片断被转发 0 次，1 次，2 次，…，h 次的概率趋向于平均，即 $1/(h+1)$，且 $\sum_{j=0}^{h}N_{(i-j+M) \bmod M}\approx W\times\frac{h+1}{M}$，由此并结合(4)式可得，当 $h\to+\infty$ 时，有

$$\lim_{h\to+\infty}\sqrt{\frac{1}{M}\sum_{i=0}^{M-1}\left(\sum_{j=0}^{h}N_{(i-j+M) \bmod M}\times\frac{1}{h+1}-N\right)^2}=\lim_{h\to+\infty}\sqrt{\frac{1}{M}\sum_{i=0}^{M-1}\left(\frac{W\times(h+1)}{M}\times\frac{1}{h+1}-N\right)^2}=\lim_{h\to+\infty}\sqrt{\frac{1}{M}\sum_{i=0}^{M-1}(N-N)^2}=0 \tag{5}$$

因此，增大(4)式中 h 的值，可以降低各节点收到的工作片断数量的标准差，提高负载均衡能力。在实际网络中，h 值

维普资讯 http://www.cqvip.com

取得过高，将导致工作片断频繁被转发，通信量增加，影响性能，并且，由于节点等待队列的存在，无限制提高 h 并不会使网络达到理想化的负载均衡状态。第 4.4 节的实验给出了 h 值与负载均衡能力的关系。

2.6.2 语义队列

在上述实现中，衡量等待队列长度的单位是队列中工作片断的数量。判断队列是否已满的标准，是排队中的工作片断的数量是否达到队列长度的上限。由前所述，节点之间存在异构性，而且任务发出者会根据网络状况动态调整工作片断的计算量，因此每个工作片断的计算量并不相同。进而每个工作片断的计算耗时是不固定的，仅根据工作片断数量来确定节点是否过载并不能达到完全公平的效果。为了解决这个问题，本文提出了语义队列的概念。

语义队列的长度用时间来表示，而不是通常的工作片断数量。这个时间是节点完成当前队列中所有工作片断所要花费的时间。在多任务操作系统中，节点在执行计算的同时，还要受系统中其它运算任务的影响，因此节点的计算能力会发生波动。为了计算当前语义队列的长度，节点实时监测当前的运算速度，并进一步导出当前队列中所有工作片断的总计算时间，即语义队列的长度。为了防止大计算量的工作片断反复因语义队列满而被拒绝，允许任务发出者在被拒绝后重传的工作片断的请求报文中指定该工作片断不受语义队列影响。节点收到“不受语义队列影响”的工作片断后，不得因语义队列满而拒绝或重定向该工作片断。引入语义队列后，为了控制节点负载，只要控制语义队列的长度，但要保证队列至少可容纳一个工作片断。这样，即使节点之间存在很大的异构性，由于队列长度是按计算耗时而不是工作片断的绝对数量来衡量的，计算力较弱的节点也不会因计算负担过重而过载，整个网络的负载均衡能力得到了进一步的提升。

3 一个分布式大质数测试系统的实现

本节将基于以上研究成果设计并实现一个分布式大质数测试系统，以检验上述方案的可行性与有效性。在第 4 节中，将进一步测试系统的各项性能。

3.1 分布式质数测试算法

设计这个分布式大质数测试系统的目的是检验上述结构化 P2P 网络上的计算力资源整合方法，因此此处不准备对各种质数测试算法的优劣进行对比，也不对质数测试算法本身进行数学上的优化，而直接使用简单试除法，即用[3,$\lceil\sqrt{N}\rceil$]区间内的所有奇数试除 N（N 为大于 3 的奇数）。若区间内的某一整数 P 除尽了 N，则 P 就是 N 的因子，N 为合数；若区间内的整数都无法除尽 N，则 N 为质数。为了使用第 2 节中的方法整合网络中计算机的闲置计算力，需要调整上述算法以适应分布式计算。首要问题是计算任务的分解与结果的汇总。这里，我们简单地把整个除数区间分成若干个小区间，并将各个小区间的除法任务作为工作片断发往网络中的节点进行计算。节点返回的计算结果有两种：一是小区间内的除数均无法除尽被除数，二是小区间的某个整数 P 除尽了被除数。若所有工作片断的计算结果都是第一种情况，则被测整数是质数；若某一个小区间的计算结果为第二种情况，则被测整数是合数。通过动态调整小区间的长度，任务发出者可以调整每个工作片断的计算量。

3.2 客户端与服务端的设计

本文设计的大质数测试系统以经典的结构化 P2P 算法 Chord 为基础，采用第 2 节的研究成果开发。结构化 P2P 网络中每个节点既是客户端也是服务端，为了叙述上的方便，此处将发出计算任务的节点称为客户端，将接受工作片断完成计算的节点称为服务端。

客户端的功能是动态地将计算任务分解成适应当前网络状况的计算量不等的工作片断，并按第 2.2-2.5 节描述的方法发往 P2P 网络中的节点。该工作片断同时被放置于客户端的未决结果队列中，等待回复。客户端的运行过程可用图 1 所示的状态转换图来表示。

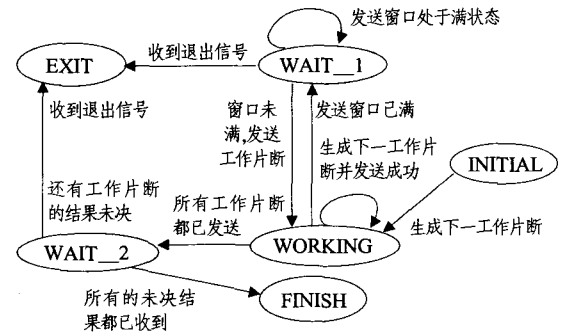


图 1 客户端状态转换图

客户端启动计算任务后，就进入图 1 所示的工作流程中。如果客户端收到一个计算结果，表明被测数不是质数时，上述过程立即终止（即发送退出信号），并输出结果。在 WORKING 状态，只要发送窗口未满，程序就根据当前的网络状态生成适当计算量的工作片断，发往网络进行计算。在 WAIT_1 状态，发送窗口处于满状态，程序等待直到窗口大小允许发送新的工作片断。在 WAIT_2 状态，所有的工作片断都已经发送，程序等待所有计算结果都返回后退出。为了处理工作片断被拒绝或超时的情况，还有一个单独的线程，定时检查未决结果队列中的工作片断。若有被拒绝或超时，则适时重传。

服务端的功能是接收工作片断并完成计算，每个服务端可设置多个独立的工作线程，同时处理收到的工作片断。在多 CPU 的计算机上，操作系统可将每个线程调度到不同的 CPU 上运行，提高单机计算力资源的利用率。每个线程的工作流程可用图 2 所示的状态转换图来表示。

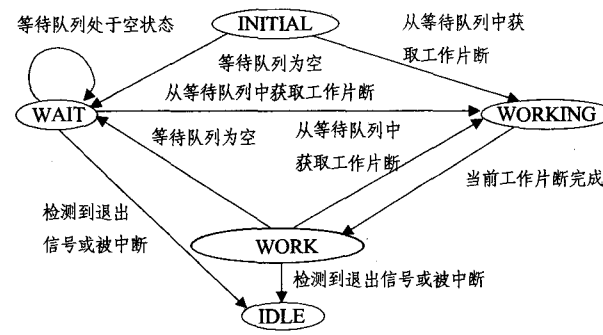


图 2 服务端状态转换图

服务端启动后开始接收客户端发出的工作片断并置于等待队列中，每个工作线程独立地从等待队列中获取工作片断，执行计算任务。程序对等待队列进行了同步，以保证多个线程同时访问时不会出现冲突。在 WORKING 状态，程序对工作片断代表的计算任务进行计算，并将结果返回给客户端。

维普资讯 http://www.cqvip.com

为了使程序工作时减小对系统其它进程的影响，当工作线程进入 WORKING 状态后，线程优先级自动降到最低。在 WAIT 状态，等待队列为空，即没有收到新的工作片断，程序将一直等待，直到有新的工作片断到达或被中止。

4 性能测试与分析

本节将结合大质数测试系统，对本文提出的基于结构化 P2P 网络的计算力资源整合方法的各项性能进行实验分析，以验证这一方案的可行性和有效性。所有性能测试是在 SPIS[8] 系统上完成的。SPIS 系统是由 P2P 研究小组自主研发的面向服务的结构化 P2P 网络支撑平台，在这一平台可以支撑各种基于结构化 P2P 网络的应用服务。本文在这一平台上实现了上述分布式大质数测试系统，并在该系统上对这一方法的性能进行了测试与分析。

4.1 对通信开销的控制能力

此项实验系检验系统控制网络通信开销的能力。实验中对 1 个长度为 70bit 的质数进行测试，参与测试的节点共有 5 个。这些节点上同时还在运行其它的任务，因此计算能力是动态变化的。计算 TT 值时使用的参数 k 为 0.1，控制网络通信时间所占比重 T 的上限 u 设定为 0.25，下限 v 设定为 0.05，即目标是将通信开销控制在 5%到 25%的区间内。客户端每收到一个工作片断，记录当时的 TT 值和 CT 值，单位为秒。TT 值与 CT 值曲线如图 3 所示。

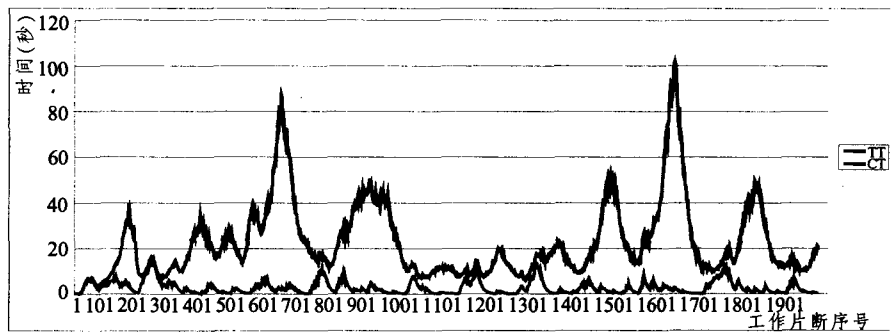


图 3 TT 值与 CT 值对比曲线

图 3 中反映出，由于结构化 P2P 网络的复杂性以及节点的动态性，不同的工作片断从发出、分配、等待、计算到结果返回的代价(即 TT 值)处在不断波动的状态中，但是通过动态调整工作片断的计算量，通信开销所占的比重并不太大。为了更好地观察通信开销占总开销的比重，实验中以 5%为单位统计了通信开销在不同范围内的工作片断的数量，如图 4 所示。

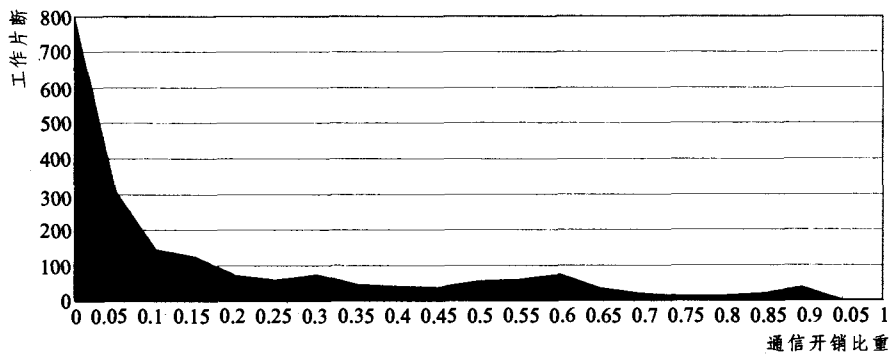


图 4 不同通信开销的工作片断数量

图 4 反映了大部分工作片断的通信开销集中于 0～15%的区间内，与控制目标 5%到 25%是相符的。对数据进一步测算后得出，所有报文的总通信开销是 12.1%，同样在预期的控制目标内。相对于结构化 P2P 网络的动态性及其路由算法的复杂性，这一数据是可以接受的。这一系列实验结果表明，第 2.3 节提出的控制通信开销的算法可以有效地控制网络通信开销的比重，提高对网络总体计算力的利用率。

4.2 动态调整工作片断发送速率的能力

此项实验系检验第 2.5 节提出的动态调整工作片断发送速率的能力。实验过程与第 4.1 节相同。实验中记录了每个工作片断的发送时戳，并以 100s 为区间单位统计发出的工作片断数量。图 5 给出了各时间段内发出工作片断的数量。

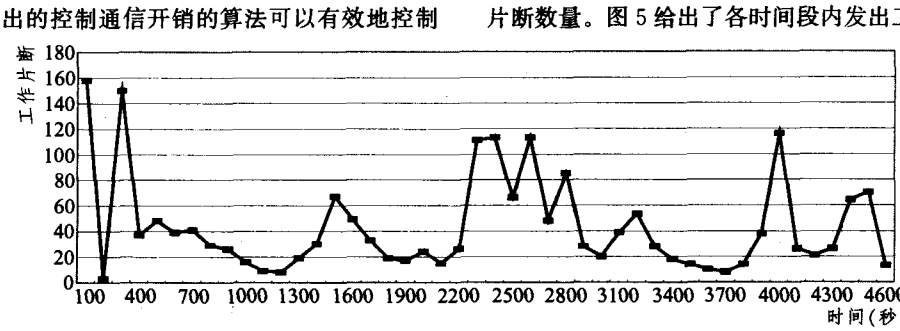


图 5 不同时间内发出的工作片断的数量

维普资讯 http://www.cqvip.com

图 5 显示出不同时间段内发出的工作片断数量具有很大的波动。分析其原因，主要有两方面：一是节点的动态性引起网络总体计算力的不断变化，二是各个工作片断的计算量也在动态地变化。这个图反映出第 2.5 节中提出的算法能够依据当前的网络状况实时地调整工作片断的发送速率，而不是以某一固定的速率发送，可以更好地拟合网络的动态变化，充分利用网络总体计算力，也避免了给工作节点带来过重的负担。第 4.3 节进一步通过实验验证了系统适应网络总体计算力变化的能力。

4.3 对网络总体计算力变化的适应能力

此项实验是检验系统适应网络总体计算力变化的能力。实验中选取了 4 个长度不同的质数进行测试，4 个质数分别是：28405810648960169，852469546242554531，3174309817-9322324351，1008331732565760816359。每个质数测试 4 次，每次使用数量不同的节点参与计算，并记录计算所耗费的时间。4 个节点的 CPU 均为 Pentium IV 3.0(HT)。得到节点数量与计算耗时的关系如图 6 所示。

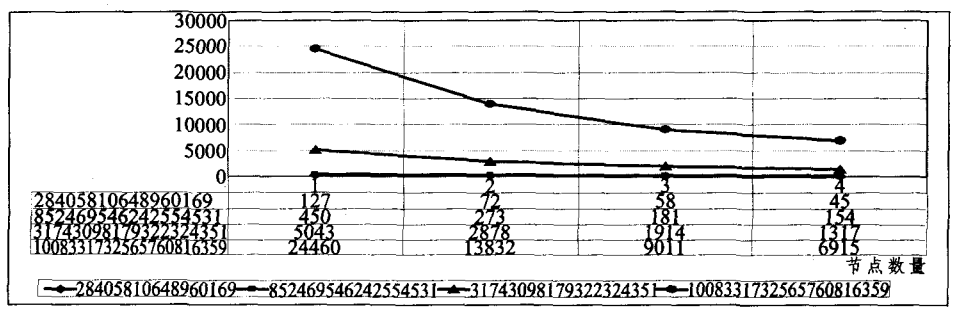


图 6 节点数量与计算耗时的关系

本文的方法将一个大的计算任务分配到网络中不同的节点上，目标是尽可能利用节点的闲散计算力，因此参与的节点越多，完成同一个任务耗费的时间应越少。以图 6 对质数 3174309817932232435 1 的实验数据为例，当节点数量为 1，2，3 和 4 时，完成整个计算过程耗费的时间分别是 5043s，2878s，1947s 和 1317s，耗费的时间呈递减趋势，这与我们的预期目标是相符合的。同样，对其它 3 个长度不同的质数的测试结果也是符合预期目标的。从以上实验数据看出，随着节点数量的增长，程序能够自动适应计算力的变化，对于同一项计算任务，计算耗时随着节点数量的增多而减少。这一结果表明，上述方案有效利用了网络的总体计算力，能够适应网络总体计算力的变化，适合在动态变化的结构化 P2P 网络上部署。

4.4 负载均衡能力

此项实验对质数 228881279557273747 进行测试，检验系统的负载均衡能力，同时观察跳数限制 h 与负载均衡能力的关系。为了避免动态调整工作片断计算量带来的误差，实验中将此功能关闭。这样，由于每个工作片断的计算量是相同的，每个节点处理的实际工作片断数量即为其负载。实验中，客户端把整个除数区间分成 2148 个工作片断，由随机产生的 15 个节点进行处理。实验共进行 7 次，跳数限制 h 的值从 0 到 6 变化。记录每次实验中各节点处理的工作片断数量，分析跳数限制 h 与工作片断数量分布之间的关系。15 个节点在 Chord 环上的分布如图 7 所示。从图中可以看出，节点在环上的分布是不均匀的。测得各节点处理的工作片断的数量分布如图 8 所示。

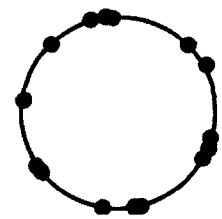


图 7 实验节点在 Chord 环上的分布

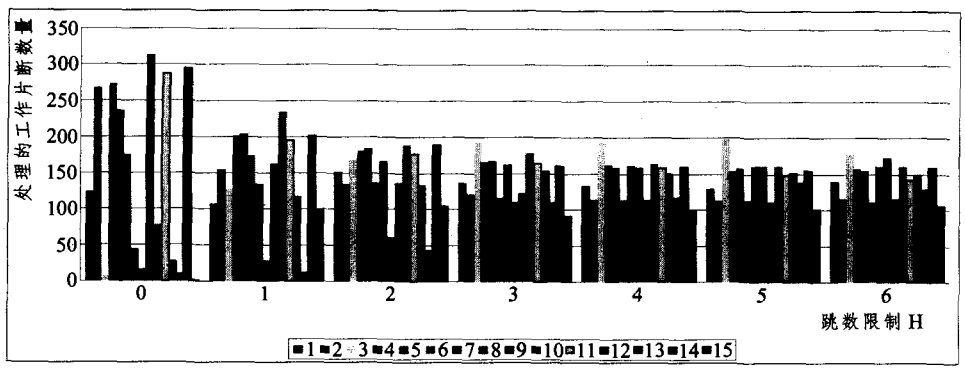


图 8 跳数限制 h 与各节点处理的工作片断数量的关系

图 8 中，横坐标是跳数限制 h，共有 7 个不同的值；纵坐标是每个节点处理的工作片断的数量。对应于每一个 h 值，分别用柱状图表示各个节点处理的工作片断的数量。从图中可以观察到，当 h 的值为 0 时(即不做工作片断重定向)，各个节点处理的工作片断数量差别很大，工作片断在节点上的分布与节点在 Chord 环上的位置有关，节点与其前驱的距离越大，处理的工作片断越多。当 h 的值逐渐增大时，工作片断在

(下转第 59 页)

维普资讯 http://www.cqvip.com

改进，使得 GEAR 路由的安全性大大增强。但同时，在安全和能耗的平衡点上，还存在着许多值得探讨的问题，很多理论和实际研究工作仍亟待解决。不得不说，研究一种高效可行的安全路由技术，是一项十分具有挑战性和富有意义的任务。

参考文献

[1] Avancha S, Undercoffer J, Joshi A, et al. Secure Sensor Networks for Perimeter Protection. Computer Networks, 2003, 43 (4): 421-435

[2] Deng J, Han R, Mishra S. INSENS: Intrusion-Tolerant Routing in Wireless Sensor Networks//The 23rd IEEE International Conference on Distributed Computing Systems (ICDCS 2003). Providence, RI, 2003

[3] De S, Qiao C M, Wu H Y. Meshed multipath routing with selective forwarding: an efficient strategy in wireless sensor networks. Computer Networks, 2003, 15(4): 481-497

[4] Yin C Q, Huang S Y, Su P C, et al. Secure Routing for Large-scale Wireless Sensor Networks//Proceedings of ICCT' 2003. Beijing: Beijing Univ Posts & Telecommun Press, 2003: 1282-1286

[5] Deng J, Han R, Mishra S. INTRSN: Intrusion-Tolerant Routing in Wireless Sensor Networks//The 23rd IEEE International Conference on Distributed Computing Systems (ICDCS 2003). Providence, RI, 2003. http://www.cens.ucla.edu/sensys03/proceedings/p324-tanachaiwiwat.pdf

[6] Tanachaiwiwat S, Dave P, Bhindwale R, et al. Secure Locations: Routing on Trust and Isolating Compromised Sensors in Location-Aware Sensor Networks

[7] Tanachaiwiwat S, Dave P, Bhindwale R, et al. Location-centric Isolation of Misbehavior and Trust Routing in Energy-constrained sensor networks//IEEE Workshop on Energy Efficient Wireless Communications and Networks (EWCN04). Phoenix, Arizona, 2004

[8] Yu Y, Govindan R, Estrin D. Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks. Tech Rep. Los Angeles, California: UCLA Computer Science Department, 2001

[9] 王建新，张亚男，王伟平，等. 移动自组网中基于声誉机制的安全路由协议设计与分析. 电子学报，2005，33 (4)：596-601

[10] Liu D G, Ning P. Location-based pairwise key establishments for static sensor networks//Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks. New York: Association for Computing Machinery, 2003: 72-82

(上接第 23 页)

各节点上的分布趋于均匀，负载平衡能力获得了提高。为了进一步定量地观察这一结果，将各节点处理的工作片断数量标准差与 h 值绘图比较，如图 9 所示。

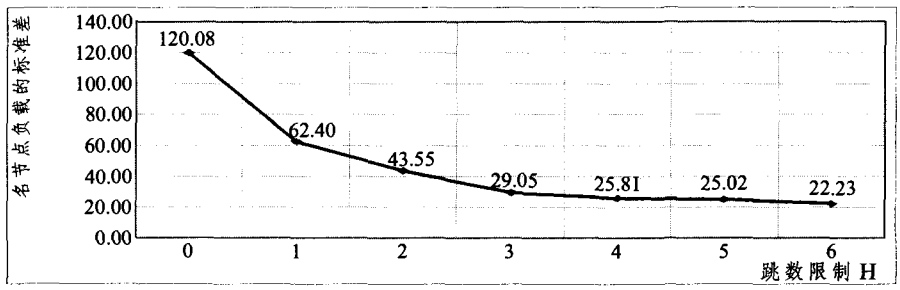


图 9 跳数限制 h 与各节点负载的标准差的关系

从图 9 中观察到，当 h 的值为 0 时，标准差达到 120.08，工作片断在节点上的分布很不均匀，没有产生负载均衡效果。随着 h 值的增大，标准差值逐步减小。当 h 值增大到 6 时，标准差值降低到了 22.23，只有 h 值为 0 时的 1/5 左右。说明工作片断重定向起到了负载均衡的效果。另一方面，上述实验结果也表明，h 值进一步增大时，对提高负载均衡能力的帮助逐渐变小。考虑到工作片断重定向带来的额外的通信开销，在实际应用中，只要根据需要选取一个合适的 h 值即可。

结束语 本文在分析了结构化 P2P 网络在计算力资源整合应用上的优势后，深入研究了基于结构化 P2P 网络的计算力资源整合的方法，并设计了基于该方法的分布式大质数测试系统。针对结构化 P2P 网络特征，设计了控制网络通信开销、控制发送速率和提高负载均衡能力的方法。实验结果表明，这一方法有效地整合了网络中的闲置计算力，并在各项系统性能测试中都有良好的表现，达到了预期的研究目的。在今后的工作中，我们将继续就这一课题进行研究，考虑设计一个通用的计算力资源整合系统，为各种不同的计算任务提供一个统一的基于结构化 P2P 网络的计算力资源整合框架。同时，对结构化 P2P 网络在其它领域的应用进行探索，推动结构化 P2P 网络的普及。

参考文献

[1] Stoica I, Morris R, Liben-Nowell D, et al. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications [J]. IEEE/ACM Transactions on Networking, 2003, 11(1): 17-32

[2] Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems[C]// Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms, Middleware 2001. Heidelberg, Springer, 2001: 329-350

[3] Zhao B, Kubiatowicz J, Joseph A. Tapestry: An infrastructure for fault-tolerant wide-area location and routing [R]. Berkeley: University of California at Berkeley, Computer Science Department, 2001

[4] Ratansamy S, FrancisP, Handnley M, et al. A scalable content-addressable network[C] // Proceedings of the 2001 ACM SIGCOMM Conference. New York: ACM Press, 2001: 161-172

[5] Malkhi D, Naor M, Ratajczak D. Viceroy: A scalable and dynamic emulation of the butterfly[C]//Proceedings of the 21st Annual ACM Symposium on Principles of Distributed Computing. New York: ACM Press, 2002: 183-192

[6] Stevens W R. TCP/IP Illustrated Volume 1: The protocols (English reprint edition) [M]. Beijing: China Machine Press, 2004: 310-312

[7] Ratnasamy S, Stocia I, Shenker S. Routing algorithms for DHTs: Some open questions[C]//Proceedings of the First International Workshop on Peer-to-peer Systems, IPTPS 2002. Cambridge, MA, USA: Springer-verlag, 2002: 45-52

[8] 东南大学计算机网络和信息集成教育部重点实验室 P2P 研究小组. SPIS 系统详细设计报告[R]. 2006

[9] Calvert L, Griffioen J, Mullins C, et al. Concast: Design and Implementation of an Active Network Service [J]. IEEE Journal on Selected Area in Communications, 2001, 19(3): 426-437

[10] Anderson P, Cobb J, Korpela E, et al. SETI@home: an experiment in public-resource computing[J]. Communications of the ACM, 2002, 45(11): 56-61

[11] Foster I, Kesselman C, Faufmann M. The Grid: Blue Print for a New Computing Infrastructure[M], 1999

[12] Dabek F, Brunskill E, Kaashoek M, et al. Building Peer-to-Peer Systems with Chord, a Distributed Lookup Service[C]// Proceedings of the 8th Workshop on Hot Topics in Operating Systems, HotOS-VIII. Schloss Elmau: IEEE Computer Society Press, 2001: 81-86