

# 兴趣和蚁群相结合的非结构化 P2P 搜索算法

李国荣<sup>1</sup>, 刘方爱<sup>1,2</sup>

(1. 山东师范大学 信息科学与工程学院, 山东 济南 250014;

2. 山东师范大学 信息技术中心, 山东 济南 250014)

**摘要:**针对传统 flooding 搜索算法面临的效率低下和网络流量过大等问题,提出了一种改进的基于兴趣和蚁群相结合的资源搜索算法(IASA)。该算法将 TTL 机制加以改进,并将兴趣相似度引入到蚁群算法的路径概率选择中,同时结合信息素的多样性和正反馈机制,积累历史搜索经验,获得路由指引信息,从而有效地指导查询请求消息的转发,将请求消息尽量发往资源可能存在的节点上。仿真实验表明:该算法能有效地指导资源搜索的方向,提高查询命中率,减少冗余消息包,其整体搜索效果较好。

**关键词:**P2P; 兴趣相似度; 蚁群算法; 信息素; 路径选择概率

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2012)07-0067-04

## Resource Search Algorithm Based on Combination of Interest and Ant Colony in Unstructured P2P Network

LI Guo-rong<sup>1</sup>, LIU Fang-ai<sup>1,2</sup>

(1. School of Information Science & Engineering, Shandong Normal University,  
Jinan 250014, China;

2. Information Technology Center, Shandong Normal University, Jinan 250014, China)

**Abstract:**In unstructured P2P network, the traditional flooding search algorithm suffers from some disadvantages of inefficiencies and excessive network traffic. To address the problem, present a resource search algorithm based on the combination of interest and ant colony (IASA). In this algorithm, the TTL mechanism was improved and interest similarity was introduced into the choice of the routing path in ant colony algorithm, and combining with the diversity of the pheromones, positive feedback mechanism was conducive to accumulate history experience and get routing guiding information, so as to effectively guide query information's forwarding, so that the query information can be sent to the proper nodes with requested resources as much as possible. Simulation results indicated that the algorithm can effectively guide the search direction and improve inquires and reduce redundancy information. On the whole, the search effect performs better.

**Key words:**P2P; interest similarity; ant colony algorithm; pheromone; path selection probability

### 0 引言

现有的非结构化 P2P 网络<sup>[1]</sup>中的搜索算法可分为盲目搜索和信息搜索两大类。盲目搜索不依赖任何已知信息,只是洪泛地遍历邻居节点搜索所需资源,如 Gnutella<sup>[2]</sup>协议使用的 flooding<sup>[3]</sup>。信息搜索算法依据自身已有的邻居节点和资源等先知信息,有目的地选

择可能性较大的邻居节点转发查询以提高发现目的资源的效率。文献[4]利用时间局部性即用户的搜索请求往往与历史搜索有较大的相关性进行资源搜索;文献[5~7]利用空间上的局部性即拥有相似资源的节点,更有可能处理彼此的查询请求,建立了基于节点局部兴趣的搜索机制。文献[8]将人工智能领域的蚁群算法引入 P2P 资源搜索,使得请求节点更加智能地选择下一跳路由。

文中受到时间和空间局部性原理的启发,在文献[9]的基础上加以改进,将蚁群算法中的信息素看作历史查询记录,并将其与节点兴趣结合起来,利用蚂蚁信息素正反馈机制来指导节点的选择查询,从而更加智能地选择下一跳路由,高效避免盲目转发消息,以减少查询所带来的网络流量,提高查询成功的概率。

收稿日期:2011-12-14;修回日期:2012-03-20

基金项目:国家自然科学基金资助项目(90612003);山东省自然科学基金资助项目(Y2007G11);济南市高校院所自主创新项目(2010 040072)

作者简介:李国荣(1986-),女,山东日照人,硕士研究生,CCF 会员,研究方向为对等网络、网格计算;刘方爱,教授,博士生导师,研究领域为并行处理、互联网络、网格计算等。

### 1 蚁群算法原理

蚁群算法<sup>[10]</sup>是一种寻找优化路径的机率型算法,一般用来解决旅行商问题。蚂蚁个体之间通过能随时间流失慢慢稀释的信息素<sup>[11]</sup>相互交流通信。每只蚂蚁前进途中都要靠感知的信息素强度指引方向,当周围没有信息素时,蚂蚁会按照自己的原有惯性运动下去,盲目的搜索食物。短路径的蚂蚁在同样的时间内来回的次数要比长路径多,从而留下更多的信息素以吸引更多的蚂蚁,同时当走这条路径的蚂蚁增多的时候,信息素又会有一个增量,最终越来越多的蚂蚁聚集到较短的路径上来。蚂蚁的这种集体行为激励了集体意识,能够启发式的促使较短的路径优先被选择。这种智能的行为比较适合 P2P 这种动态网络,但为避免局限于局部最优解引起的网络拥塞,文中对 TTL 机制进行了改进,并增加了拥塞广播消息。

### 2 IASA 资源搜索算法

#### 2.1 兴趣相关度

P2P 网络中每个节点所拥有的资源文档共同表现了节点兴趣。文档的兴趣可以用 k-高频词即文档中一定数量的高频词及其分布模式来表示。首先对文档用分词算法<sup>[12]</sup>进行分词处理,得到关键词集合  $K = \{k_1, k_2, \dots, k_i, \dots, k_n\}$ ;统计每个关键词在该集中的词频,得到词频向量  $F = (f_1, f_2, \dots, f_i, \dots, f_n)$ ;计算向量  $F$  中元素的第  $k$  个最大值  $f_{kmax}$ ,把  $f_i < f_{kmax}$  的元素置为 0,得到该节点的 k-高频词向量  $F^k$ 。k-高频词向量能有效地代表节点的兴趣喜好。

对于两个 k-高频词向量分别为  $F_i^k$  和  $F_j^k$  的节点  $i$  和  $j$  来说,它们之间的兴趣相似度的计算公式为:

$$\text{Sim}(i, j) = \frac{F_i^k \cdot F_j^k}{|F_i^k| \times |F_j^k|} \quad (1)$$

Sim 的值越大,表示节点  $i$  和  $j$  的兴趣相关程度越高,反之表示兴趣相关程度低。

#### 2.2 IASA 改进的资源搜索算法

##### 2.2.1 IASA 网络模型

在 IASA 资源搜索算法里,整个 P2P 网络是一个由很多蚁巢以及连接蚁巢的通路组成的小世界网络,其中每个对等节点相当于存放着一定共享资源的蚁巢,这些资源体现节点的兴趣偏好,并可以被网络上其他对等节点访问。在 P2P 网络中搜索资源的过程和蚂蚁觅食的行为类似:申请源节点发出的查询请求消息包视作蚂蚁,请求的资源看作食物,食物源是存放目标资源的节点,搜索经过的节点序列看作是路径。蚂蚁初期搜索时,环境中没有有用的信息素,使得寻径有一定的盲目性,因此在转发查询消息前借助节点的兴趣偏好事先判断目标资源更可能存在哪些节点上,使

蚂蚁的走向智能化,在某种程度上能够提高算法的搜索效率。

##### 2.2.2 IASA 算法数据结构

为支持 IASA 算法在 P2P 资源搜索中的应用,每个节点需要维护一些不同类型的数据结构。

1)本地资源表 R 存储本节点共享资源的索引信息和本节点的当前状态,便于其他对等节点了解本节点的访问。

2)热点文件缓存表 H 缓存最近一段时间被访问的资源地址信息,可以直接从该表中获取经常被访问的资源的 URL,而减少不必要的搜索开销。H 表的维护更新可采用 LRU 算法。

3)信息素表  $\tau$  用一个  $M \times N$  的二元矩阵表示,其中,  $M$  被用来表示资源特征的数量,也就是通过分词处理后所得到的高频关键词;  $N$  被用来表示与当前节点互为邻居节点的个数。元素  $\tau_{ku}$  代表由节点  $i$  到邻居节点  $u$  的路径上关于关键词  $k$  的信息素浓度。网络搜索的初始阶段矩阵中所有元素都设为最小值 0.1。

4)邻居兴趣表  $\eta$  是一个  $n \times 2$  的矩阵,其中的  $n$  表示当前节点有多少个邻居节点,  $\eta_{ui}$  表示当前节点  $i$  到其邻居节点  $u$  的距离,这里用从  $i$  发送一个蚂蚁到  $u$  所用时间  $T_{iu}$  的倒数表示。  $\eta_{u2}$  表示用公式(1) 计算出来的当前节点  $i$  和邻居节点  $u$  的兴趣相似度。

##### 2.2.3 路径选取概率

申请源节点生成搜索蚂蚁选择邻居节点开始搜索资源的过程中,用户输入的关键词可能不是唯一的,但是节点的搜索请求在某种程度上体现了该节点的相关兴趣,因此针对蚁群算法收敛速度慢等缺点,在计算路径选取概率时,将节点的兴趣参与进来,使得路径消耗最小且兴趣相关度大的节点优先被寻径蚂蚁搜索到。每个邻居节点的选取概率用如下改进的公式计算:

$$P_{key,j}^k = \frac{\tau_{key,j}^\alpha \eta_{j1}^\beta \eta_{j2}^\theta}{\sum_{j \in \text{allowed}_k} \tau_{key,j}^\alpha \eta_{j1}^\beta \eta_{j2}^\theta}, j \in \text{allowed}_k \quad (2)$$

其中:  $\tau_{key,j}$  为邻居节点  $j$  关于本次查询消息中关键字信息素的浓度值,  $\alpha$  为信息素浓度在搜寻策略中所占的比重,  $s$  为当前节点  $i$  的邻居节点中蚂蚁  $k$  尚未访问的节点。  $\eta_{j1}$ 、 $\eta_{j2}$  分别为当前节点  $i$  和邻居节点  $j$  的距离和兴趣相似度。  $\beta$ 、 $\theta$  分别为两节点距离和兴趣相似度在路径选择中的权重。

为提高查询效率,设定一个路径选取概率门限  $q \in (0, 1)$ ,选择满足  $P_{key,j} \geq q$  且无任何拥塞广播消息的邻居进行查找。若  $P_{key,j}$  都小于  $q$ ,则选取其中比较空闲且值较大的 top-k<sup>[13]</sup> 个节点路径进行查找。

##### 2.2.4 信息素更新机制

为反映信息素的时效性避免其无限制的增加,在

一定时间间隔内采用式(3)对网络中每个对等节点的信息素执行挥发操作。

$$\tau_i = (1 - \rho)\tau_i \quad (3)$$

其中,  $\rho$  为信息素的挥发因子, 响应蚂蚁在返回时需更新经过的节点序列上关于 key 的信息素浓度。为更好地指引未来搜索蚂蚁查找对该类资源, 对原有的信息素更新机制进行改进, 使得花费时间越短且找到资源越多的节点信息素增量越大, 改进的信息素更新公式为

$$\tau_{(key,j)} = (1 - \rho)\tau_{(key,j)} + \Delta\tau_{(key,j)} \quad (4)$$

其中,

$$\Delta\tau_{(key,j)} = \begin{cases} \gamma \frac{S_{find}}{S_{max}} + (1 - \gamma) \frac{TTL_{max}}{TTL}, & \text{蚂蚁经过点 } j \\ 0, & \text{其它} \end{cases}$$

式中: 信息素的挥发因子  $\rho \in (0, 1)$ ;  $\Delta\tau_{(key,j)}$  是本次查询中节点  $j$  的关于 key 的信息素变化量, 初始时, 对  $\forall j, \Delta\tau_{(key,j)} = 0$ 。  $S_{find}$  表示已经搜索到的资源的数量,  $TTL$  表示搜索蚂蚁到节点  $j$  所用的时间,  $\gamma$  表示资源匹配率的权重值。

### 2.2.5 TTL 改进机制

对等网络在进行资源搜索的过程中, 只是简单的用 TTL 即路由跳数表示搜索消息的生命周期, 但当网络中出现拥塞的时候, 可能搜索消息一直不能到达下一跳, 这时搜索消息不了解网络状态一直在搜索, 从而不但搜索不到资源还加重了网络负担, 所以文中搜索蚂蚁的生命周期用 TTL 和时间戳 TS 同时控制, 当  $TTL = 0$  或者达到 TS 的时候, 搜索蚂蚁都会死亡。同时如果节点网络负载较重, 就会向周围邻居节点广播拥塞消息, 这样可以避免蚁群算法收敛于局部最优。改进的 TTL 机制更能真实反映网络状态, 从而减少不必要的网络流量。

### 2.2.6 IASA 资源搜索步骤

本节将主要针对 IASA 资源搜索算法步骤展开讨论, 其具体步骤如下:

Step1: 当请求节点  $i$  搜索关键词为 key 的资源时, 先检查  $i$  的本地资源表 R, 若有资源满足条件, 则将其直接返回, 否则再查找热点文件缓存表 H, 若找到则直接与相应的地址沟通下载资源, 在此过程中用变量  $S_{find}$  统计已找到的资源数目。若查询结果已达到预先设定的上限  $S_{max}$ , 则算法终止并退出, 否则继续执行 Step2。

Step2: 请求节点  $i$  释放  $ANT_{search}$  消息的过程中同时侦听来自周围邻居节点的拥塞广播信息, 摒弃负载过重的邻居, 组成比较空闲的可选邻居集合  $Z$ , 对  $\forall j \in Z$ , 根据  $i$  与  $j$  的兴趣相关度  $Sim(i, j)$  以及信息素浓度  $\tau_{key,j}$ , 来计算邻居  $j$  的路径选择概率  $P_{key,j}^k$ , 选择  $P_{key,j}^k \geq q$

的邻居节点分别复制消息进行搜索。如果没有满足条件的节点, 则选择搜索目前比较空闲且概率值较大的 top-k 个节点。每只查询蚂蚁都需维护一个禁忌表  $ban(ANT_{search})$ , 以此来记录查找过程中被访问的节点。为防止无限制搜索, 每只蚂蚁都要设置一个生存周期, 该生存周期由 TTL 和时间戳 TS 共同决定, 当  $TTL = TTL_{max}$  时或者搜索时间超过 TS, 搜索蚂蚁自动死亡。

Step3: 当搜索蚂蚁到达下一跳邻居节点  $n$  后, 先检查禁忌表判断自己的兄弟蚂蚁是否已访问过该节点, 若已被访问, 则选择与  $n$  的兴趣相关度最大且未被访问的拥挤度不大的邻居节点进行搜索。否则, 搜索表 R 和 H 是否含有目标资源, 如果有, 则生成携带本节点的相关信息响应蚂蚁  $ANT_{reply}$ , 按照  $ban(ANT_{search})$  表中记录的原搜索路径向申请源节点  $i$  返回命中消息并更新路径上各节点的关于关键字 key 的信息素表。当请求节点收到响应信息时, 在缓冲区表 H 中缓存该资源的地址, 并相应修改本地资源 R 表, 若缓冲区已满, 则用 LRU 算法进行替换。响应消息到达源节点后, 源节点查看变量  $S_{find}$ , 若  $S_{find} \geq S_{max}$  则调用该资源。此时, 判断搜索时间是否超过时间戳 TS, 若没超过则搜索蚂蚁的 TTL 减 1, 若此时 TTL 为 0, 则销毁搜索蚂蚁, 否则, 返回 Step2。

Step4: 如果查询蚂蚁在时间戳内在节点  $n$  没有找到合适的资源且  $TTL < TTL_{max}$ , 则对路径选取概率  $P_{key,j}^k$  进行计算, 搜索拥挤程度比较小的合适路径直到时间戳已到或者  $TTL = TTL_{max}$  为止。当一个对等节点被访问后, 应修改禁忌表对其做相应的访问标记, 以避免兄弟蚂蚁重复查找, 从而减少不必要的搜索开销。

## 3 实验仿真

为了验证算法的有效性, 实验采用 PeerSim 模拟器<sup>[14]</sup>对 IASA 资源搜索算法进行仿真。PeerSim 是一个开源的、基于 java 开发的大型 P2P 模拟器框架, 简单易用、可扩展性强, 可支持上百万个节点的模拟仿真。

### 3.1 仿真场景模拟

实验中采用均匀随机图模型作为网络拓扑, 设置 P2P 规模为 1000 个网络对等节点, 网络中节点采用小世界模型分布, 节点的平均邻居数为 10, 每个节点使用的时钟周期一样, 随机选择 100 个节点存放查询资源, 将每个节点的资源数固定为 5, 随机产生 10 个资源查询请求。整个 P2P 网络中含有 10 个不同兴趣种类的资源, 对等节点中都可以随机分配到 6 个以内的不同兴趣资源, 同时, 规定不允许未分配现象的出现, 这些不同兴趣的资源存放着数量不等的兴趣资源信息。最终, 对该 IASA 实验的参数设置如表 1 所述:

表 1 IASA 参数设置

实验参数	取值	说明
$\alpha$	0.6	信息素在搜索策略中所占的比重
$\beta$	0.2	启发函数在概率选择中的权重
$\theta$	0.2	兴趣相似度在路径选择中的重要性
$q$	0.6	Ant 选取路径时的概率门限值
$\rho$	0.1	信息素挥发系数
$TTL_{max}$	20	蚂蚁最大的生存时间
$S_{max}$	65	搜索资源的期望值
$\gamma$	0.55	匹配资源数量与查询时间的比值
$\tau_{min}$	0.1	信息素表初始阶段最小值
TS	100	搜索消息的时间戳

### 3.2 实验结果及分析

#### 3.2.1 平均搜索成功率

图 1 反映了 IASA 算法同 flooding 算法、k-random (设  $k=6$ ) 算法的平均搜索成功率随着 TTL 值变化的情况,从图 1 中可以看出,随着 TTL 值的增加,3 种算法的搜索成功率都在不断变大,但是, flooding 算法和 k-random 算法搜索的盲目性使得其搜索命中率随时间的改变不会有明显的变化。而 IASA 算法运用蚁群算法的信息双反馈和改进的 TTL 机制,优先选择可能性较大的节点转发,更容易找到目标,搜索的盲目性的有效避免,使其平均搜索成功率一直维持在一个较高的状态。当  $TTL=8$  时,整个网络的对等节点慢慢对全局信息和局部信息有了一定的学习和认知,借助这些动态变化的信息,其搜索成功率开始达到一个相对稳定的状态。

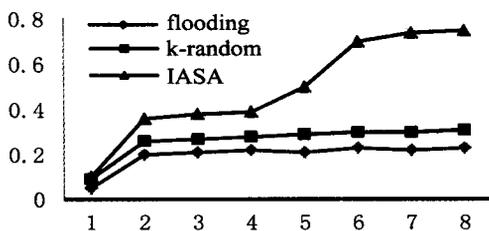


图 1 搜索成功率与 TTL 之间的关系

#### 3.2.2 平均发送消息数

平均发送消息数是指在保证相同搜索效果的前提下发送的平均消息包数量,算法性能见图 2。

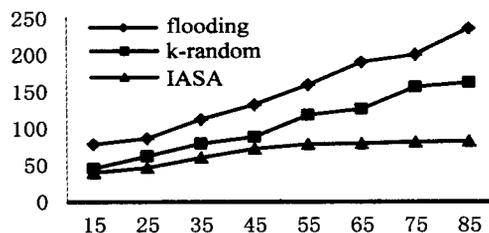


图 2 发送查询消息数同搜索资源数之间的关系

从图中可以看出,在搜索效果相同的条件下,相对于 flooding 算法和 k-random 算法,IASA 算法平均发送的搜索消息数明显减少。IASA 算法在搜索资源的过

程中不断学习,并根据先知信息更新节点的信息素值,从而选择资源可能存在的搜索路径,更智能地查找目标资源,从而有效避免冗余。

#### 3.2.3 平均响应时间

从用户角度考虑,查询请求的响应时间也是衡量 P2P 检索效率的重要指标,实验结果见图 3。

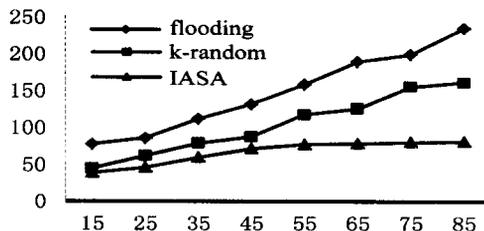


图 3 响应时间同搜索资源数之间的关系

图 3 中可以看到在非结构化 P2P 网络中要达到相同的搜索效果,k-random 算法由于随机选择转发响应时间最大,而 flooding 算法的响应时间最小。flooding 算法由于不加选择的向其周围所有的邻居节点洪泛转发查询消息,响应时间较小,能保证查全率,但是大量的冗余消息包容易造成网络拥塞。而 IASA 算法在信息素和兴趣相关度的指导下,能更具针对性的用较小的响应时间搜索到目标资源。

由实验结果可知,IASA 资源搜索策略在搜索成功率、平均消息发送数量和平均响应时间方面相比 flooding 算法和 k-random 算法具有较大优势。

## 4 结束语

针对非结构化 P2P 网络中传统 flooding 算法搜索效率低和冗余消息多等缺点,文中模拟蚂蚁觅食行为提出一种基于兴趣和蚁群相结合的资源搜索算法,将兴趣相似度引入蚁群算法的路径选取,使得查询请求消息优先转发到最有可能存在目标资源的节点上,有效避免了冗余消息所带来的网络流量。同时 TTL 和时间戳的相互作用,更能真实反映网络状态,从而减少不必要的开销。为进一步改善搜索算法,如何均衡负载,避免热点文件被过度访问是下一步研究的重点。

#### 参考文献:

- [1] 张春红,袁晓峰,弭伟,等. P2P 技术全面解析[M]. 北京:人民邮电出版社,2010.
- [2] Gnutella Home Page[EB/OL]. 2010. <http://www.gnutella.com/>.
- [3] Jiang Hongbo, Jin Shuding. Exploiting dynamic querying like flooding techniques in unstructured peer-to-peer networks[C]//Proceedings of 13th IEEE Internet Conference on Network Protocol. [s.l.]:[s.n.],2005.
- [4] 秦宏伟,代亚非,李晓明. 针对访问成功率的 P2P 动态网络

⑧-6 FPGA 中,建立系统的硬件平台,然后在 xmd 命令窗口里键入如下命令,将编译好的 U-Boot 文件下载到 SP605 评估板的 DDR RAM 中:

- (1) connect mb mdm
- (2) down U-Boot

如果下载没有错误,则在 xmd 命令窗口里键入 run 命令,运行 U-Boot 代码。如果能从串口输出正确的启动信息,就表明 U-Boot 移植基本成功,文中所移植的 U-Boot 程序在 SP605 评估板上的运行情况如图 3 所示。

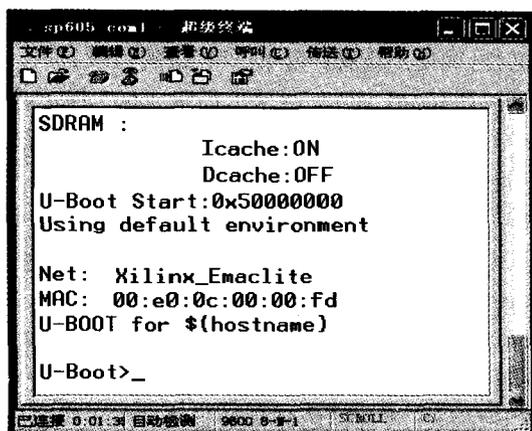


图 3 运行在 SP605 评估板上的 U-Boot 程序

U-Boot 运行后,可以通过它提供的命令测试目标板的一些基本情况,如检查内存是否读写正常、网络通信是否正常。此外用户也可以添加自定义的功能,将 U-Boot 作为主控程序,应用到某些对软件系统需求不高的地方,完成简单的嵌入式系统的控制。

## 5 结束语

U-Boot 是一个功能强大的 Bootloader 开发软件,

适用的 CPU 平台及支持的嵌入式操作系统很多。文中是笔者在实际开发过程中根据相关资料进行摸索,并在成功移植了 U-Boot 的基础上总结出来的。对于不同的 CPU 和开发板,其基本的方法和步骤是相同的,希望能对相关嵌入式系统的设计人员有所帮助。

### 参考文献:

- [1] 华清远见嵌入式培训中心. 嵌入式 Linux 应用程序开发标准教程[M]. 第 2 版. 北京:人民邮电出版社,2009.
- [2] Šimek M. Embedded Operating System for MicroBlaze [J]. Acta Polytechnica,2006,46:37-39.
- [3] Bmyninckx H, LeuvenK U. Real-time and Embedded Guide [J]. Mechanical Engineering,2001,56:5-9.
- [4] 张进,姜威. U-boot 的启动流程及移植[J]. 国外电子元件,2005(5):11-14.
- [5] 徐亚鹏,谢凯年. 用 U-Boot 构建 IXP2350 目标系统的引导程序[J]. 计算机技术与发展,2007,17(5):9-14.
- [6] Xilinx Company. SP605 Evaluation Platform[M]. [s. l.]:Xilinx Company,2007.
- [7] Xilinx Company. MicroBlaze Processor Reference Guide[M]. [s. l.]:Xilinx Company,2009.
- [8] 叶肇晋. 基于 XILINX FPGA 片上嵌入式系统的用户 IP 开发[M]. 西安:西安电子科技大学出版社,2008.
- [9] 赵峰. FPGA 上的嵌入式系统设计实例[M]. 西安:西安电子科技大学出版社,2008.
- [10] 杨强浩. 基于 EDK 的 FPGA 嵌入式系统开发[M]. 北京:机械工业出版社,2008.
- [11] 周涛,刘有源. 基于 MIPS 的 Au1500 NC 板上的启动程序 U-Boot 设计[J]. 计算机技术与发展,2006,16(8):94-96.
- [12] 吴桂清,张峰铭,秦丽,等. 基于 MicroBlaze 的振动信号采集系统的设计与实现[J]. 电子设计工程,2010(3):37-39.

(上接第 70 页)

- 对象定位模型[J]. 软件学报,2005,16(5):894-902.
- [5] 刘业,杨鹏. 基于自组织聚类的结构化 P2P 语义路由改进算法[J]. 软件学报,2006,17(2):339-348.
- [6] 杨静. 基于用户兴趣的 P2P 搜索机制研究[D]. 西安:西安电子科技大学,2010.
- [7] 朱晓姝,周娅,黄桂敏. 基于小世界层次分布式路由模型研究[J]. 计算机工程,2006,32(15):120-122.
- [8] 宋雪梅. 蚁群算法的改进及应用研究[D]. 唐山:河北理工大学,2006.
- [9] 王新生,李学,贾冬艳. 基于蚁群算法的非结构化 P2P 资源搜索机制[J]. 计算机工程,2009,35(7):189-194.
- [10] Bontoux B, Feillet D. Ant colony optimization for the traveling purchaser problem [J]. Computer & Operations Research, 2008,35(2):628-637.
- [11] 蓝慧琴,钟诚,李智. 一种基于蚁群算法的非结构化 P2P 网络搜索算法[J]. 计算机技术与发展,2006,16(10):26-28.
- [12] 青桂仙,苏筱蔚,陈淑艳. 中文文本挖掘的无词典分词的算法及其应用[J]. 吉林工程学院学报,2002,23(1):16-18.
- [13] 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.
- [14] PeerSim [EB/OL]. 2010. <http://peersim.sourceforge.net/>.