

# Performance Improvement of Hint-based Locating & Routing Mechanism in P2P File-sharing Systems<sup>1</sup>

Hairong Jin Shanping Li Gang Peng Tianchi Ma

College of Computer Science, Zhejiang University, Hangzhou, P.R.China 310027  
hrjin@eyou.com shan@cs.zju.edu.cn e\_pglmary@hotmail.com mtc@cad.zju.edu.cn

**Abstract.** Hint-based Locating & Routing Mechanism (HBLR) derives from the locating & routing mechanism in Freenet. HBLR uses file location hint to enhance the performance of file searching and downloading. In comparison with its ancestor, HBLR saves storage space and reduces file request latency. However, because of the inherent fallibility of hint, employing location hint naively for file locating in P2P file-sharing system will lead to under-expectant performance. In this paper, hint's uncertainty and related bad results are analyzed. According to the causation, pertinent countermeasures including credible hint, master-copy transfer, and file existence pre-detection are proposed. Simulation shows the performance of HBLR is improved by adopting the proposed policies.

## 1 Introduction

The Hint-based Locating & Routing Mechanism (HBLR) [3] is based on the *document routing model* [1] in Freenet [2]. It processes file location hints searching instead of direct file searching and distributes file location hints instead of files themselves. Comparing to Freenet, HBLR has two obvious advantages: disk space saving on peers, and file transfer time reducing by downloading file from the selected peer.

However, distributing hints instead of files also has its drawbacks. Following are several problems accompanied with proposed solutions:

1. Stale hint set. If all hints that a requestor receives are stale, the request will fail undoubtedly. However, the file may exist on some peers, most probably on the file's publisher. So, employing the publisher as a credible file holder, and keeping the corresponding hint on peers can provide at least one downloading source.

2. Credible file holder absence. If a publisher encounters space shortage, it may delete previously published files. Moving the file to another peer, and maintaining necessary links between the old and new holders can resolve the problem.

3. Repetitious downloading attempts. If the hints pointing to some relatively better positions are all stale, the requestor will fail to download several times until a correct hint related position in its turn. It's inefficient and a file existence pre-detection mechanism is needed.

---

<sup>1</sup> The Project Supported by Zhejiang Provincial Natural Science Foundation of China (No.602032)

The rest of this paper is organized as follows: Section 2 describes the proposed mechanisms. Section 3 gives simulation and discussion. Section 4 gives some related work. Finally, the author's work is summarized at Section 5.

## 2 Improving Hints' Accuracy in HBLR

The searching and distribution methods for file in Freenet and for hint in HBLR are equal. Therewith, their searching success ratios are almost equal too. If the hint's accuracy can be ensured, HBLR's performance can be improved.

### 2.1 Credible Hint and Master-Copy

For a given file, if (1) there is a credible file holder, (2) hint about this holder is kept on all the peers, which have the hint entries about the given file, and (3) the hint about the credible holder will always be transferred to the requestor, then HBLR's request success ratio can be improved to Freenet's level.

A publisher, which provides the file for share, will keep the file as long as possible. So, the publisher is a quite credible holder, condition (1) can be satisfied. In *finite-hint* [3] solution, every hint items are transferred back naturally. In *full-hint* [3] solution, as long as a special policy is adopted, assuring the publisher related hint to be transferred back, condition (3) can be satisfied. A peer in HBLR obtains hints during three procedures: (a) Publishing; (b) Hints transferring in request; (c) Updating after file downloading. No matter in which way, so long as condition (3) is satisfied, the holder related hint could always be kept. So, condition (2) can be satisfied.

We call the publisher *credible holder*, and the publisher related hint *credible hint*. The copy of a file on its publisher is called *master-copy*.

### 2.2 Master-Copy Transfer

The file's publisher is a quite credible holder, but not a completely credible holder. When it encounters space shortage, it may delete a previously published file. The publisher should transfer the master-copy to a neighbor peer. The criteria in target peer selection include free space and communication latency. The publisher also remains a link pointing to the new holder for master-copy access. The new holder is also enabled to transfer the file while facing space shortage.

### 2.3 File Existence Pre-Detection

After receiving hints, the requestor selects a best peer for downloading. But, if the file on the best peer was deleted, the downloading attempt fails, and reselecting is needed. Though with credible hint and master-copy transferring mechanism, the file will be obtained in the end, repetitious attempts are inefficient. File existence pre-detection should be involved in the selection step, and file existence becomes a precondition.

### 3 Simulations

The success ratio, space usage, and average file service time were concerned in simulations. 100 peers and 4000 different files were involved. The files' average size was variable and each peer contributed 800Mbytes storage space to Freenet or HBLR.

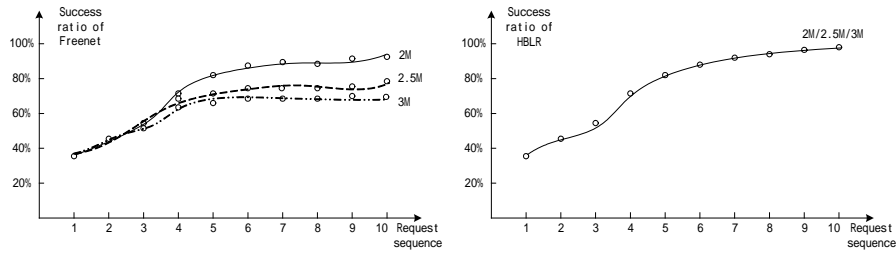


Fig. 1. Success ratio of file requests

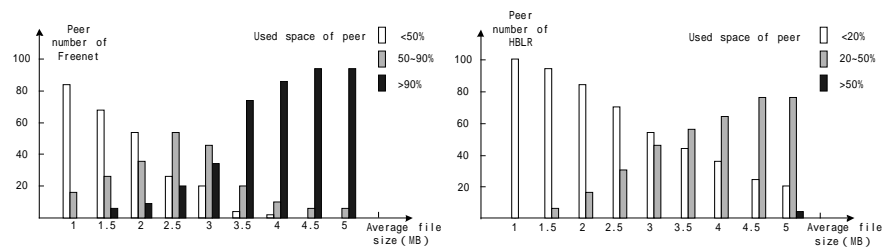


Fig. 2. Space usage

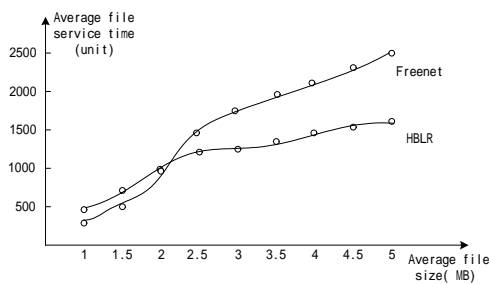


Fig. 3. File service time

Success ratios of requests were compared. The two systems both processed ten continuous request sequences, each containing 1000 requests. The files' average size was adjusted among 2M/2.5M/3Mbytes. As Fig.1 shows, HBLR achieved approximate success ratio to Freenet. When file size became bigger, HBLR's success ratio exceeded that of Freenet in latter sequences. The cause was that Freenet deleted some cached copies for providing space to the new coming, thus reduced copy distribution.

Space usage was seriously concerned. The simulation process included 3000 file requests. The files' average size was adjusted from 1Mbytes to 5Mbytes. As Fig.2 shows, Freenet used much more space than HBLR, especially with bigger files.

As Fig.3 shows, HBLR paid out for its relatively complex operation, its performance of average file service time was below Freenet with smaller files. But it did better when files were bigger. Because the bigger the files were the less copies distributed in Freenet.

According to the simulations, HBLR with credible hint and master-copy transferring mechanism behaved well.

## 4 Related Work

The improving work was based on HBLR [3], which derived from Freenet [2]. The basic locating models in the systems are all document routing model [1].

Cooperation is an essential character and requirement in P2P. Other cooperative systems' experience can be referenced for building P2P cooperative mechanisms. P. Sarkar and J. H. Hartman proposed a cooperative caching mechanism, using master copy and hint [4]. Michael D. Dahlin et al developed another cooperative caching model to improving distributed file system performance [5]. In their models, useful copies are forwarded to some other places when facing local deletion.

## 5 Conclusions and Future Work

In this paper, credible hint and master-copy transferring mechanism were imported to resolve hint uncertainty problem in HBLR. These new elements performed energetically in HBLR, improved its performance.

HBLR is an ongoing project and some work needs to be done in the future. The credible hint is a minimum step for improving hint's accuracy. Advanced hint management mechanism is still needed.

## Reference

1. Dejan S. Milojicic et al. Peer-to-Peer Computing. Hewlett-Packard Internal Document, HPL-2002-57. March 2002.
2. Ian Clarke et al. Freenet: A Distributed Anonymous Information Storage and Retrieval System. Lecture Notes in Computer Science. 2000.
3. Hairong Jin, Shanping Li, Tianchi Ma, and Liang Qian. A Hint-based Locating & Routing Mechanism in Peer-to-Peer File Sharing Systems. GCC2003, December 2003.
4. Prasenjit Sarkar and John H. Hartman. Hint-Based Cooperative Caching. ACM Transactions on Computer Systems, Vol. 18, No. 4, Pages 387-419. Nov. 2000.
5. Michael D. Dahlin et al. Cooperative Caching: Using Remote Client Memory to Improve File System Performance. The First Symposium on Operating Systems Design and Implementation, 1994.