Tracker vs. DHT Performance Comparison for P2P Streaming
draft-hu-ppsp-tracker-dht-performance-comparison-01.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on March 20, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.
Abstract

The draft makes performance analysis in two kinds of resource exchange and discovery methods, tracker based and DHT based architectures in P2P streaming. Our analysis shows that centralized tracker solution for resource discovery (and chunk lookup) has much shorter response time than highly distributed DHT approach.
Table of Contents

1. Introduction.....................................................4
2. Resource discovery.............................................4
   2.1. Lookup efficiency.......................................4
   2.2. Network traffic..........................................5
   2.3. Host requirement........................................6
3. Chunk discovery................................................7
   3.1. Lookup efficiency.......................................8
   3.2. Network traffic..........................................8
   3.3. Host requirement........................................9
4. Conclusion......................................................11
5. References........................................................12
   5.1. Normative References....................................12
   5.2. Informative References..................................12
1. Introduction

Assume there are D resources shared by N peers in a P2P streaming system, there are different kinds of methods for a given peer to locate or discover a specific resource [PPSP]. One is tracker-based method, where a peer reports the resource(s) it has to a centralized server (i.e. the tracker). When the tracker receives a resource request from a peer, it returns to the requesting peer with a list of peers which have the requested resource. Another method is a DHT-based (such as [Chord]), fully-distributed lookup protocol, where resource information is stored by all the peers in the P2P network. We estimate the performance of the two methods.

For P2P streaming usage scenarios, N is the number of active users in a P2P streaming software (such as PPLive, PPStream), D is the number of channels (for live streaming) or videos (for VoD) the software provides. For a popular P2P streaming software, there are about 10 million active users and about 100 thousand resources.

2. Resource discovery

P2P streaming is usually divided into chunks. A chunk is a basic unit of partitioned streaming, which is used for the purpose of storage and advertising to neighbors what parts of a movie a peer holds [Sigcomm:P2P streaming]. In this part of comparison, we only consider the discovery of the coarse information (resource information), and compare the discovery of the grain information (chunk information) in the next part.

In other words, we compare the following two methods. In tracker-based method, the tracker stores the resource information (given a resource Ri, the list of peers [Pi] that have Ri), while the chunk information (Pi has which chunks) is exchanged using peer gossip. Similarly, in DHT-based method, only resource information is obtained using DHT method, and chunk information is also exchanged using peer gossip after the requesting peer gets the peer list that has the requested resource. One assumption in DHT-based method is that all the nodes are geographically distributed on the whole Internet.

2.1. Lookup efficiency

In the estimation, we assume average RTT in the network is 200ms, N = 10,000,000 and D = 100,000.

In tracker-based method, assume one tracker is used, resource information is stored using dictionary: {Ri: [Pi]}, then:
(a) Lookup message: O(1)
(b) Lookup operations: O(1)

Lookup operation is to find one record in N records in the tracker.

(c) Lookup latency: O(1)*RTT = 200ms

The delay caused by lookup operations is negligible compared with the network delay.

In DHT-based method:

(a) Lookup message: O(log(N)) = 23
(b) Lookup operations: log(N)*O(1) = 23

Each node maintains information about O(log(N)) other nodes. Lookup operation is done by routing through O(log(N)) other nodes toward the destination. In each node, lookup operation is to find one record in O(log(N)) records.

(c) Lookup latency: O(log(N))*RTT = 23*200ms = 4.6s

The delay caused by lookup operations is negligible compared with the network delay.

Summary: Tracker-based method is much faster than DHT-based method. The 4.6s lookup latency is relatively high in P2P streaming applications.

2.2. Network traffic

Assume on average each peer requests new channels or videos every T seconds, then totally there are N/T requests per second in the network. Assume on average the size of a message is S Bytes.

In the following estimation, we assume T = 60 sec. The size of the response message is usually larger than the request message. For simplicity, we assume S = 1K Bytes.

In tracker-based method:
(a) Number of messages per second: N/T*2 = 10,000,000/60*2 = 3.3*100,000
There are $N/T$ channel/video requests per second in the network, and each request takes 2 messages (one request and one response).

(b) Size of messages per second: $N/T \cdot 2 \cdot S = 3.3 \cdot 100,000 \cdot 1K = 3.3 \cdot 100,000,000 = 0.33GBytes$

(c) Number of messages in node join/leave: $O(1)$

In DHT-based method:

(a) Number of messages per second: $N/T \cdot 2 \cdot \log(N) = \frac{10,000,000}{60} \cdot 2 \cdot 23 = 7.7 \cdot 1,000,000$

There are $N/T$ channel/video requests per second in the network, and each request takes $2 \cdot \log(N)$ messages.

(b) Size of messages per second: $N/T \cdot 2 \cdot \log(N) \cdot S = 7.7 \cdot 1,000,000 \cdot 1K = 7.7 \cdot 1,000,000,000 = 7.7GBytes$

(c) Number of messages in node join/leave: $O((\log N) \cdot (\log N)) = 541$

Summary: Tracker-based method has much smaller network traffic overhead than DHT-based method, but both methods are acceptable in P2P streaming applications.

2.3. Host requirement

In tracker-based method:

(a) Memory requirement:

Information of $D$ resources is stored in the tracker. If on average one peer has $C$ resources (the user watches the video or stores the video for other peers), then the average length of the peer list for each resource is $N \cdot C / D$. Assume each peer is represented by $P$ Bytes, then the total size of this table is $D \cdot (N \cdot C / D) \cdot P = N \cdot C \cdot P$ Bytes.

Assume $C = 10$, $P = 20$ Bytes, then memory requirement is $N \cdot C \cdot P = 10,000,000 \cdot 10 \cdot 20 = 2GBytes$

(b) Number of received requests per second: $N/T = \frac{10,000,000}{60} = 1.67 \cdot 100,000$

Assume on average each peer requests new channels or videos every $T$ seconds, then totally there are $N/T$ requests per second to the tracker.
(c) Size of request/response messages per second: \( \frac{N}{T} \times 2 \times S = 3.33 \times 100,000 \times 1K = 0.33 \text{GBytes} \)

\( S \) is the average size of request/response message, and we assume \( S = 1K \) Bytes.

In DHT-based method:

(a) Memory requirement:

Information of \( D \) resources is stored in \( N \) peers, so on average each peer stores information for \( \frac{D}{N} = \frac{100,000}{10,000,000} = 0.01 \) resource.

For those peers that store resource information, the average length of the peer list for each resource is \( \frac{N \times C}{D} \). Assume each peer is represented by \( P = 20 \) Bytes, then the memory requirement is \( (\frac{N \times C}{D}) \times P = 10,000,000 \times 10 / 100,000 \times 20 \) Bytes = 20KBytes.

In addition, each node maintains information about \( O(\log(N)) \) other nodes, assume the size of the neighbor information is \( E \) Bytes, this part of memory requirement is \( O(\log(N)) \times E \) Bytes = 23*10 = 230 Bytes (assume \( E = 10 \) Bytes).

(b) Number of received requests per second:

There are \( \frac{N}{T} \) channel/video requests per second in the network, and each request takes \( \log(N) \) request messages. So on average the number of requests one peer receives per second is: \( \frac{N}{T} \times \log(N) / N = \log(N) / T = 23 / 60 = 0.4 \)

(c) Size of request/response messages per second: \( 2 \times \log(N) / T \times S = 2 \times 0.4 \times 1K = 0.8 \) KBytes

\( S \) is the average size of request/response message, and assume \( S = 1K \) Bytes.

Summary: DHT-based method has much less host resources requirement than tracker-based method. For robustness and performance considerations, multiple trackers can be used.

3. Chunk discovery

In this part, we compare the following two methods. In tracker-based method, the tracker stores the resource information, and the chunk information is exchanged using peer gossip. While in DHT-based method, both resource information and chunk information are obtained using
DHT method. For the DHT-based method, we use the first solution in [PPSPChunk]. For the tracker-based method, since the tracker only stores the resource information, the performance analysis is the same as the previous part. In this part, we also add the performance analysis for the peer gossip. We assume each peer gossips with M neighbors.

3.1. Lookup efficiency

In the following estimation, we assume average RTT in the network is 200ms, \( N = 10,000,000 \), \( D = 100,000 \) and \( M = 20 \).

Tracker side:

(a) Lookup message: \( O(1) \)

(b) Lookup operations: \( O(1) \)

(c) Lookup latency: \( O(1) \times \text{RTT} = 200\text{ms} \)

Peer side:

(a) Lookup message: \( M \times O(1) = 20 \)

Each peer sends a lookup message to \( M \) neighbors.

(b) Lookup operations: \( O(1) \)

(c) Lookup latency: \( O(1) \times \text{RTT} = 200\text{ms} \)

In DHT-based method:

(a) Lookup message: \( O(\log(N)) = 23 \)

(b) Lookup operations: \( \log(N) \times O(1) = 23 \)

(c) Lookup latency: \( O(\log(N)) \times \text{RTT} = 23 \times 200\text{ms} = 4.6\text{s} \)

Summary: Tracker-based method is much faster than DHT-based method. The 4.6s lookup latency is relatively high in P2P streaming applications.

3.2. Network traffic

In tracker-based method:
Tracker side:
(a) Number of messages per second: \( N/T*2 = 10,000,000/60*2 = 3.3*100,000 \)
(b) Size of messages per second: \( N/T*2*S = 3.3*100,000*1K = 3.3 *100,000,000 = 0.33GBytes \)
(c) Number of messages in node join/leave: \( O(1) \)

Peer side:
Assume each peer sends gossip message every I seconds, \( I = 10 \) sec.
(a) Number of messages per second: \( M*N/I*2 = 20*10,000,000/10*2 = 4*10,000,000 \)
(b) Size of messages per second: \( M*N/I*2*S = 4*10,000,000*1K = 40GBytes \)

In DHT-based method:
Assume the rate of the video is \( R \) KBytes/sec, the size of the chunk is \( Z \) Kbytes, then the number of chunk requested per second is \( R/Z \).
Assume \( R = 32 \) Kbytes/sec, \( Z = 16 \) Kbytes, then \( R/Z = 2 \).
(a) Number of messages per second: \( N*(R/Z)*2\log(N) = 10,000,000*2*2*23 = 1,000,000,000 \)
There are \( N*(R/Z) \) chunk requests per second in the network, and each request takes \( 2*\log(N) \) messages.
(b) Size of messages per second: \( N*(R/Z)*2\log(N)*S = 1,000,000,000*1K = 1TBytes \)
(c) Number of messages in node join/leave: \( O((\log N)*(\log N)) = 541 \)

Summary: Tracker-based method has much smaller network traffic overhead than DHT-based method, but both methods are acceptable in P2P streaming applications.

3.3. Host requirement

In tracker-based method:
Tracker side:
(a) Memory requirement: \( N*C*P = 10,000,000*10*20 = 2GBytes \)
(b) Number of received requests per second: \( N/T = 10,000,000/60 = 1.7*100,000 \)
(c) Size of request/response messages per second: N/T*2*S = 3.3*100,000*1K = 0.33GBytes

Peer side:

Assume the size of the bitmap is Bm Bytes, Bm = 1K.

(a) Memory requirement: M*Bm = 20*1K = 20KBytes

(b) Number of received requests per second: M/I = 20/10 = 2

(c) Size of request/response messages per second: M/I*2*S = 2*2*1K = 4KBytes

In DHT-based method:

(a) Memory requirement:
Assume the number of chunks in one resource is H, H = 10000. D*H chunk information is stored in N peers, so on average each peer stores information for D*H/N = 100,000*10,000/10,000,000 = 100 chunks.

The average length of the peer list for each chunk is N*C/D. Assume each peer is represented by P Bytes, then the memory requirement is (N*C/D)*P*(100 chunks) = 20KBytes*(100 chunks) = 2MBytes.

In addition, each node maintains information about O(log(N)) other nodes, this part of memory requirement is O(log(N))*E Bytes = 23*10 = 230 Bytes (assume E = 10 Bytes).

(b) Number of received requests per second:

There are N*(R/Z)=2N chunk requests per second in the network, and each request takes log(N) request messages. So on average the number of requests one peer receives per second is: 2*log(N) = 2*23 = 46

(c) Size of request/response messages per second: 2*log(N)*2*S = 46*2*1K = 92 KBytes

S is the average size of request/response message, and assume S = 1K Bytes.

Summary: DHT-based method has much less host resources requirement than tracker-based method. For robustness and performance considerations, multiple trackers can be used.
4. Conclusion

Centralized tracker solution for resource discovery (and chunk lookup) has much shorter response time than highly distributed DHT approach. In a large network, the DHT approach’s response time can be very long (on the order of seconds), which is not suitable for delay sensitive applications. On the other hand, the per-host requirement of the tracker is higher than that of DHT nodes, but is still within reach of a $1000 commodity PC. Because the lookup messages have business value and should be stored for latter analysis, using a small number of trackers will make the job much easier than a highly distributed approach.
5. References

5.1. Normative References

[1] [PPSP] Problem Statement of P2P Streaming Protocol (PPSP), Y. Zhang et al., Internet Draft
draft-zhang-ppsp-problem-statement-05.txt

[2] [Chord] Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, Ion Stoica et al., In SIGCOMM 01.

[3] [Sigcomm:P2P streaming] Challenges, Design and Analysis of a Large-scale P2P-VoD System, Yan Huang et al., In SIGCOMM 08.

[4] [PPSPhunk] Chunk Discovery for P2P Streaming, N. Zong, Internet Draft draft-zong-ppsp-chunk-discovery-00.txt

5.2. Informative References

Author’s Addresses

Yan Hu
NEC Labs China
Email: huyan@research.nec.com.cn

Yong Xia
NEC Labs China
Email: xiayong@research.nec.com.cn