2011 Seventh ACM/IEEE Symposium on Architectures for Networking and Communications Systems

Fast Content Distribution on Datacenter Networks

Shakir James and Patrick Crowley
Computer Science and Engineering
Washington University in St. Louis
{sjames, pcrowley}@wustl.edu

ABSTRACT

Peer-to-peer (P2P) applications distribute large files fast. That makes them popular on the Internet and has motivated their use on datacenter networks. On datacenter networks, however, these Internet applications waste bandwidth. To fully use available bandwidth, we propose the P2P copy (PCP) application. Results with a prototype show that PCP reduces content distribution times by an order of magnitude.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*

General Terms

Performance, Design, Experimentation

Keywords

P2P, datacenter network

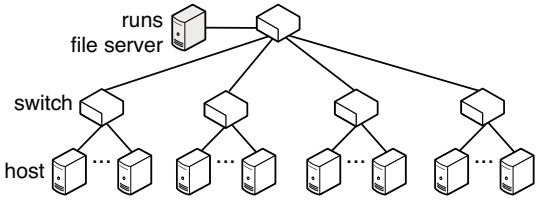
1. INTRODUCTION

Client applications download from server applications. P2P applications, however, also download from each other. For large files and many clients, P2P applications distribute files faster with fewer servers. Twitter and Facebook, for example, now use P2P applications to quickly distribute software updates to thousands of hosts. Twitter's application is open-source and uses an Internet protocol: BitTorrent.

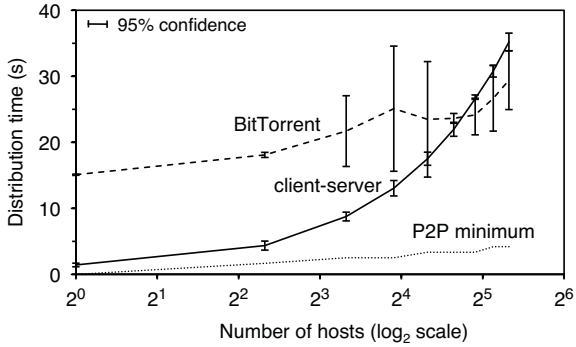
Twitter's BitTorrent application is called Murder. Compared to a central server, Murder claims to reduce distribution times from 40 minutes to 12 seconds [5]. The minimum, however, is less than a second [4, 3]. Murder fails to achieve the minimum time because it underuses available bandwidth. To fully use available bandwidth, we propose the P2P copy (PCP) application for datacenter networks.

2. MOTIVATING EXAMPLE

Datacenter networks interconnect hundreds of thousands of hosts. Yet the hosts can send and receive data simultaneously at the speed of their network interface [1, 2]. Such networks offer full available bandwidth. In practice, however, networks are oversubscribed. This reduces initial costs but also available bandwidth [1]. Applications that fully use available bandwidth achieve the minimum distribution time.



(a) Network topology.



(b) Distribution times for 100MB file.

Figure 1: Datacenter network experiments.

To test the relationship between distribution time and bandwidth usage, we ran an experiment on a network testbed. Figure 1(a) shows our network topology: a 10 gigabit, core switch interconnects the four gigabit, access switches. Each access switch interconnects a cluster of 10 hosts with gigabit network interfaces. The hosts in a cluster can communicate with each other at full rate, but the access switch's uplink to the core switch is oversubscribed.

The experiment tests the hypothesis: applications that use more available bandwidth result in lower distribution times. To that end, it uses two applications: client-server and BitTorrent. The client-server application uses bandwidth at the file server; BitTorrent also uses bandwidth at the 40 clients, so it should be faster.

Figure 1(b) compares the distribution times, the time it takes to distribute the file to $log_2 N$ hosts. Client-server's distribution times, as a function of $x = log_2 N$, grows at a rate of 2^x: it increases linearly with N as expected. But BitTorrent's distribution times seem counterintuitive: for

978-0-7695-4521-9/11 $26.00 © 2011 IEEE
DOI 10.1109/ANCS.2011.21

$N \leq 40$, it is greater than or equal to client-server's. This begs the question whether, in fact, BitTorrent mostly uses the bandwidth at the clients.

3. DESIGN OVERVIEW

We start with a clean slate. Our streamlined design uses more bandwidth because it eliminates the overhead inherent in Internet protocols such as BitTorrent. This sidesteps the question of why these protocols are inefficient on datacenter networks. Even so, fundamental differences between the Internet environment and the datacenter environment make a case for exclusion:

- *Incentive:* Internet peers may be selfish, datacenter ones are altruistic. Is an incentive mechanism necessary?
- *Availability:* Internet peers may be fickle, datacenter ones are constant. Is a part selection policy necessary?
- *Bandwidth:* Internet connections may be slow, datacenter one are fast. Is the pipelining of file parts necessary?

The proposed design assumes that such mechanisms are dispensable on datacenter networks. Simulations already show that a "naïve randomized" strategy can be asymptotically optimal [4]. Our design follows this minimalist approach.

Two design decisions are important. First, should the system's controller be distributed or centralized? A distributed controller, which runs on the network of peers, is more complicated than a centralized controller, which runs on a host. Moreover, Murder's centralized controller handles thousands of simultaneous requests [5]. Thus, a centralized controller suffices.

The other design decision relates to file distribution. There are two options: peers either push to or pull from each other. Suppose that there are N peers and M parts. A centralized, optimal schedule exists where the peers transfer one file part in discrete rounds. To distribute all file parts in the minimum time, the peers take $z = M + \lfloor log_2 N \rfloor$ rounds[4].[1]

In the pull method each peer sends a request for some file part. The peers send r_{pull} requests:

$$r_{pull} = \sum_{i=0}^{z} N = N^{M+\lfloor log_2 N \rfloor} \quad (1)$$

In the push method the peers send a request for consent. The peers send r_{push} requests:

$$r_{push} = \sum_{i=0}^{z} 2^i = 2^{(M+\lfloor log_2 N \rfloor)+1} - 1 \quad (2)$$

Therefore, the push method reduces the number of control messages from $O(N^M)$ to $O(2^M)$.

[1]This optimal assumes equal host bandwidth and full available bandwidth.

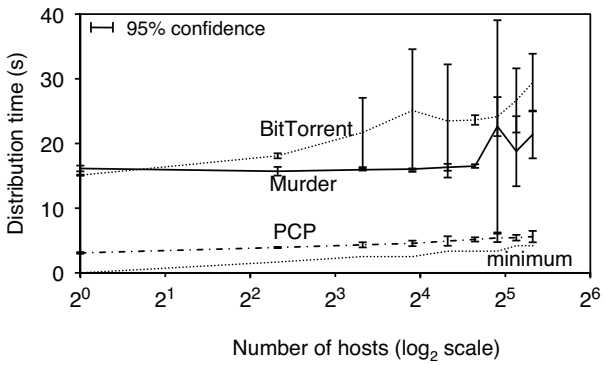


Figure 2: Distribution times for three P2P applications.

4. EXPERIMENTAL RESULTS

The goal of the evaluation is to compare the performance of P2P applications. The performance criterion is distribution time: the time to distribute a file from a source to N destinations. The testbed configuration is the same as in Figure 1(a). We vary the type of P2P application: BitTorrent's mainline application, BitTorrent; Murder; and PCP.

Figure 2 shows the results of the experiments. It shows the average of five repetitions. For $N \geq 8$ hosts, Murder's and BitTorrent's distribution times are mostly the same. PCP's distribution times, however, is significantly lower.

5. SUMMARY

Current P2P applications underuse bandwidth on datacenter networks. Our new PCP application uses more bandwidth and hence reduces content distribution times: experimental results with a prototype show an order of magnitude improvement. Further measurements will include field tests.

6. REFERENCES

[1] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, SIGCOMM '08, pages 63–74, New York, NY, USA, 2008. ACM.

[2] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. Vl2: a scalable and flexible data center network. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, SIGCOMM '09, pages 51–62, New York, NY, USA, 2009. ACM.

[3] R. Kumar and K. Ross. Peer-assisted file distribution: The minimum distribution time. In *Hot Topics in Web Systems and Technologies, 2006. HOTWEB '06. 1st IEEE Workshop on*, HOTWEB'06, pages 1 –11, Boston, Massachusetts, 2006. IEEE Computer Society.

[4] J. Mundinger, R. Weber, and G. Weiss. Optimal scheduling of peer-to-peer file dissemination. *Journal of Scheduling*, 11:105–120, April 2008.

[5] Twitter Engineering. Murder: Fast datacenter code deploys using bittorrent, July 2010.