New Insights into File Distribution in Data Center Networks: Performance Analysis of BitTorrent

Song Li*, Jun Li†, Dongsheng Wei*, Xin Wang* and Jin Zhao*
*Shanghai Key Lab of Intelligent Information Processing
School of Computer Science, Fudan University, Shanghai 200433, China
†Department of Electrical and Computer Engineering, University of Toronto, Canada

***Abstract*—In data center networks (DCN), with the extensive development of cloud computing, multicast has become an important and pervasive traffic pattern. Compared with designing specific network protocols and building corresponding functionality into switches and softwares for data center servers, using BitTorrent takes advantages of easy deployment, extensive practical uses and native scalability in multicast. We specially propose a theoretical model to analysis the finish time of multicast of BitTorrent in DCN, and use the model and simulations to compare BitTorrent with Datacast, the representative network-equipment based multicast method in DCN.**

I. Introduction

Data centers are supporting various cloud services and among the traffic produced by those inside the data center, data multicast is a pervasive traffic pattern which is supporting important functions.

To deliver data reliably, solutions can be categorized into two classes, *i.e., network-equipment assisted* and *end-host based* [1]. Though network-equipment assisted protocols, such as Datacast [2], are able to exploit the network topology of the data center and achieve good scalability, they all require the support of switches (like cache) while modern data centers are equipped with low-end commodity switches with limited capability and programmability. On the other hand, end-host based multicast protocols can fit in with data centers better as they do not require the support of switches, which is used by Facebook and Twitter for code or binaries deployment.

Consequently, there are two groups of researches. One attempts modifying BitTorrent into the application to suit for data multicast in data center networks and taking advantage of its simpleness and reliability, like [3]. The other one tries to invent new network-equipment assisted protocols because of the need for fast deployment and less traffic redundancies, aiming at achieving better performance than BitTorrent, like [1] [2], which has more cost and complexity though. However, there has been no fundamental understandings of the performance of BitTorrent in data center network. We believe that the knowledge of performance of BitTorrent in data center network helps guide the choice making of applications of file distributing in data center network.

Although BitTorrent achieves great success on the internet, it is not necessary to keep complicated components on data center, such as incentive mechanisms, the NAT/firewall traversal and *etc*, since servers in the data center are almost homogeneous and maintained by a single authority and can be jointly optimized. Thus, the BitTorrent protocol and client running in the data center can be greatly simplified and thus made light-weighted. Hence, rudely borrowing the performance analysis research on the wild BitTorrent into the data center can underestimate the performance of that in data center, and thus it is necessary to theoretically understand the BitTorrent performance specially in data center network.

In this work, we aim at finding the potential of BitTorrent performance in data center network. Instead of analyzing the wild BitTorrent protocols, we consider the the simplified and optimized general BitTorrent protocol. We believe analyzing with these features can help us fundamentally understand the performance of BitTorrent in data center networks.

II. Theoretical Analysis

Since finish time is the major concern for the data multicast performance in data center, to find the potential of BitTorrent performance, we focus on the lowerbound of finish time of BitTorrent file distribution.

BitTorrent uses Local Rarest First(LRF) as the piece selection strategy. As proved in [4], Network Coding (NC) has the minimum possible delay for any topology and the performance of LRF is very closed to that of NC. Hence, we reasonably use NC in the model to find the lowerbound of finish time.

Previous analysis of BitTorrent fails to consider the network topology, which is not accurate for data center network. In our work, we consider the topology. Among all the possible topologies for data center, we choose the most representative one, *i.e*, Fat Tree. In our model, the data center is represented as a graph $G = (V, E)$, in which V is the set of servers and E represents the path between the servers. The BitTorrent process is regarded as a *stochastic process* in the graph G, and we use Markov model to analyze.

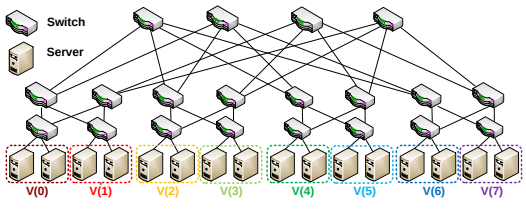

Fig. 1. A dividing example on Fat Tree where $k = 4$. Servers in Fat Tree are divided into 8 sets, namely, $V(0)$, $V(1)$, ..., $V(7)$.

In our analysis, the phases of distributing process are characterized by the status of nodes. The rate from one state to another is defined as *transition intensity*. As Fig. 1 shows, we divide the servers in the Fat Tree into numbers of sets, and the servers in the same rack belong to the same set. We define $X_t(i,j)$ as the number of nodes in set $V(i)$ who have exactly j pieces at time t, where $j = 0,1,2,...,m$, $i = 0,1,2,...,\lfloor k^2/2 \rfloor - 1$ for a k-Fat Tree, and m is the number of file pieces. All the $X_t(i,j)$ together characterize the network state, and the transition intensity of the process of set i changes from status $X_t(i,j-1)$ to $X_t(i,j)$ is denoted by $q_t(i,j)$.

We first have the transition intensity. Defining $p(i)$ as the probability the nodes of set i exist in receiver group, we have,

$$q_t(i,j) = \min(X_t(i,j-1), q'_t(i,j)), \tag{1}$$

, where

$$q'_t(i,j) = ((|V(i)| \cdot p(i) - X_t(i,0)) \cdot \frac{|V(i)| \cdot p(i) - 1}{N' - 1} + \sum_{l \neq i} \min((|V(l)| \cdot p(l) - X_t(l,0)) \cdot (1 - \frac{|V(l)| \cdot p(l) - 1}{N' - 1}), \frac{k}{2} \cdot p(l)) \cdot \frac{|V(i)| \cdot p(i)}{N' - |V(i)| \cdot p(i)}) \cdot \frac{X_t(i,j-1)}{|V(i)| \cdot p(i) - X(i,m)} \cdot \mu \tag{2}$$

In this equation, N' is calculated as follows,

$$N' = \sum_{i=0}^{\lfloor k^2/2 \rfloor - 1} |V(i)| \cdot p(i) \tag{3}$$

We define $T^{(\epsilon)}$ as the finish time at which $1-\epsilon$ of all nodes finish downloading, where $\epsilon \in [0,1]$. We can find the lowerbound[1]:

$$T^{(\epsilon)} \geq inf\{t | \forall x_t(i,m) \geq 1-\epsilon, i = 0,1,2,...,\lfloor \frac{k^2}{2} \rfloor - 1\} \tag{4}$$

in which $x_t(i,j)$ can be calculated by the ODEs:

$$\dot{x_t} = \frac{q_t(i,j) - q_t(i,j+1)}{|V(i)|}, \quad i = 0,1,2,...,\lfloor \frac{k^2}{2} \rfloor - 1, j = 0,1,2,...,m \tag{5}$$

where i is the set id, and m is the number of pieces to send.

III. Evaluation

We evaluate the finish time of BitTorrent and Datacast in Fat Tree. From Figure 2 we can see that the theoretical finish time of Datacast is smaller than that of BitTorrent. However, as Figure 3 illustrates interestingly, the gaps between the two lines gradually get smaller when the number of pieces increases. According to Figure 3, the percentage of the finish time bias becomes lower than 5% when transmitting not that large number of pieces. Hence, although the finish time of the method of Datacast is smaller than that of BitTorrent, the gap between those two method is getting sufficiently small when number of pieces increases, showing that BitTorrent on data center networks has a very close performance to Datacast.

[1]We directly give the result without proving because of the limited space.

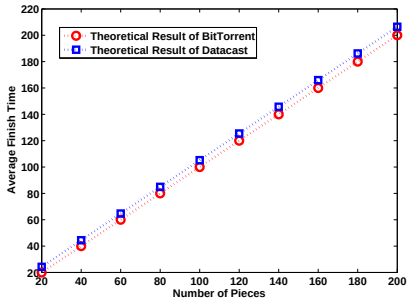

Fig. 2. A comparison between BitTorrent and Datacast on Fat Tree. The number of pieces is from 20 to 200. The number of nodes in Fat Tree is 54 with $k = 6$.

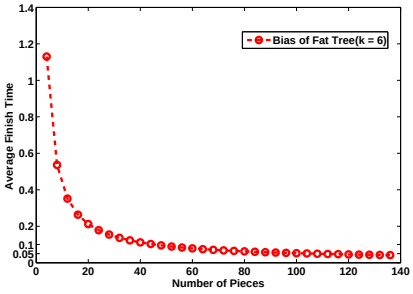

Fig. 3. The proportion of bias from BitTorrent to Datacast on Fat Tree.

IV. Conclusion

The result shows that BitTorrent file distribution can perform fast in Fat Tree. Hence, designing a novel protocol that greatly exceed the finish time of BitTorrent in Fat Tree seems to be very hard. Moreover, BitTorrent is much simpler and cheaper to implement than network-assited protocols like Datacast. Hence, for applications that need fast distribution of files in topology like Fat Tree, BitTorrent is a good choice. Further researches will focus on the other metrics of BitTorrent like network stress, background traffic, and other topologies.

Acknowledgement

This work was supported in part by the National Science Foundation of China under Grant 61171074, the National S&T Major Project of China under Grant 2010ZX03003-003-03, Program for New Century Excellent Talents in University under Grant NCET-11-0113. Xin Wang is the corresponding author.

References

[1] D. Li, M. Xu, M. chen Zhao, C. Guo, Y. Zhang, and M.-Y. Wu, "RDCM: Reliable Data Center Multicast," in *Proc. of IEEE INFOCOM*, 2011.

[2] J. Cao, C. Guo, G. Lu, Y. Xiong, Y. Zheng, Y. Zhang, Y. Zhu, and C. Chen, "Datacast: a scalable and efficient reliable group data delivery service for data centers," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, ser. CoNEXT '12, 2012, pp. 37–48.

[3] M. Chowdhury, M. Zaharia, J. Ma, M. I. Jordan, and I. Stoica, "Managing Data Transfers in Computer Clusters with Orchestra," in *Proc. of ACM SIGCOMM*, 2011.

[4] D. Niu and B. Li, "Topological Properties Affect the Power of Network Coding in Decentralized Broadcast," in *Proc. of IEEE INFOCOM*, 2010.