Bootstrapping Large-scale DHT Networks

Jae Woo Lee[†], Henning Schulzrinne[†], Wolfgang Kellerer[‡], Zoran Despotovic[‡]
[†]Department of Computer Science, Columbia University, New York, USA
{jae,hgs}@cs.columbia.edu
[‡]DoCoMo Communications Laboratories Europe, Munich, Germany
{kellerer,despotovic}@docomolab-euro.com

ABSTRACT

The recent disruption of the Skype voice-over-IP system, triggered by a massive reboot of the hosts around the world, brought to light the importance of efficient bootstrapping in a large-scale peer-to-peer network. Thus far, the problem of bootstrapping a DHT network from a near-total failure has received limited attention from the research community. We present an outline of our plan to investigate DHT bootstrapping mechanisms for large-scale deployments on the Internet.

1. INTRODUCTION

On August 16th 2007, the Skype peer-to-peer network suffered a worldwide failure, which left millions of users unable to use the popular voice-over-IP service for two days. The disruption was later attributed to a previously unseen bug in the Skype client software, which prevented the rapid formation of the overlay network when a large fraction of the host computers rebooted at the same time after receiving a routine operating system update [1]. Although this incident was caused by a software bug, therefore is not an indication of a fundamental difficulty in a peer-to-peer system, it does illustrate an important point: bootstrapping a large-scale peer-to-peer network from scratch or after a near-total failure is not an academic exercise, but a real-world problem that merits careful consideration when designing and deploying such systems.

In recent years, structured peer-to-peer overlay networks based on distributed hash tables (DHT) became popular as the substrates on which global-scale distributed systems are built, and the problem of maintaining such overlay networks in the presence of churn has been the subject of much academic research. The problem of *bootstrapping* such systems, however, has not received as much attention. There have been a number of proposals of constructing specific DHT networks efficiently from scratch [9, 4, 8, 6], but there has not been an effort to compare different DHT bootstrapping mechanisms in order to predict their performance in the real world deployment scenario similar in scale to Skype network, which is claimed to have millions of simultaneous nodes. Moreover, the previous proposals commonly base their algorithms on the existence of an unstructured overlay network where each node possesses a handful of pointers to other modes in the network, but does not specify how such a network is initially formed in the real world.

In this abstract, we outline the elements of our upcoming research effort to study DHT bootstrapping. Although the effort is only in a nascent stage, some important requirements and directions have been identified. The remainder of this abstract is organized as follows. Section 2 describes some of the parameters and methods of our planned approach. Section 3 introduces our ideas on using multicast-based service discovery techniques to enhance the bootstrapping performance of DHT networks.

2. ANALYSIS FRAMEWORK

2.1 System Model Assumption

We will identify the underlying system model assumptions of the existing bootstrapping proposals and investigate their validity when applied to the current Internet. We currently believe that all existing proposals start with a common system model called a *knowledge graph* [5], a weakly connected graph modeling a collection of nodes each of which has a handful of pointers to other nodes in the network. Jelasity, *et al.* [7] provide valuable insights on the properties of the knowledge graphs formed by different flavors of gossiping algorithms. We plan to investigate if such algorithms are applicable to the current Internet, and explore other mechanisms to form knowledge graphs on the Internet.

*CoNEXT'*07, December 10-13, 2007, New York, NY, U.S.A.

2.2 Algorithms and Topologies

We plan to compare different bootstrapping algorithms leading to different DHT topologies. There are multiple axes to consider. A bootstrapping algorithm can be evaluated by the total number of messages sent, the message sizes, and the time required to form the target DHT. The baseline for comparison is the naive algorithm of having all the nodes join the overlay sequentially. The algorithm's resilience to churn is also important since it is reasonable to assume that the underlying knowledge graph is in a state of flux.

Different bootstrapping algorithms target different DHT topologies such as ring [9] or prefix tree [4, 8]. We will compare the algorithms for their relative performances within each target topology, but we will also compare them across topologies in order to discover any inherent bootstrapping characteristic that might be present only in certain topologies.

We will examine various ways to employ parallelism. Hierarchical DHT designs are likely to make parallel bootstrapping easier.

2.3 Measurement Parameters

We should be careful to choose a performance metric that can be translated to the real-world user experience. The message count used in many papers is not adequate as it does not take into account the latency associated with the underlying physical network.

3. AUGMENTING DHT WITH MULTICAST

We plan to pursue a number of ideas related to using multicast-based service discovery mechanisms such as Zeroconf [2] to augment DHT overlay networks. This line of thinking was inspired by our previous work where we built a software tool that can connect multiple Zeroconf networks using DHT [3]. The focus in this effort, however, will be to examine whether the multicast mechanisms can be used to enhance the system's bootstrapping performance.

Multicast can be used to locate the peer nodes in the local link participating in the same overlay network. Furthermore, an overlay network can employ the multicast-based super-node architecture where only a single representative in a multicast domain participates in the global overlay.

Another interesting variation to use multicast is to treat the subnet as a node in a DHT overlay, rather than the individual hosts. For example, the IDs in the DHT can be based on the network addresses, rather than the host IP addresses. In this scheme, all hosts in a local subnet share a single ID, and have the identical routing table. The overlay routing is done at the subnet level, rather than at the individual host level. Once a request reaches the destination subnet, a multicast-based discovery method such as Zeroconf is used to resolve the request to the responsible host. We also need a way to deliver a request to a subnet. Since one cannot open a TCP connection to a network, each routing table entry will contain a sampling of a fixed number of host IP addresses along with the network address. A host from that sample is randomly selected as the recipient of a request. Since the request is further multicasted in the target subnet, any host in the network can receive the request from outside.

4. REFERENCES

[1] What happened on August 16. `http://heartbeat.skype.com/2007/08/what_happened_on_august_16.html`.

[2] Zero Configuration Networking. `http://www.zeroconf.org/`.

[3] Zeroconf-to-Zeroconf Toolkit (z2z). `http://sourceforge.net/projects/z2z/`.

[4] K. Aberer, A. Datta, M. Hauswirth, and R. Schmidt. Indexing data-oriented overlay networks. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 685–696. VLDB Endowment, 2005.

[5] I. Abraham and D. Dolev. Asynchronous resource discovery. In *PODC '03: Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pages 143–150, New York, NY, USA, 2003. ACM.

[6] D. Angluin, J. Aspnes, J. Chen, Y. Wu, and Y. Yin. Fast construction of overlay networks. In *SPAA '05: Proceedings of the seventeenth annual ACM symposium on Parallelism in algorithms and architectures*, pages 145–154, New York, NY, USA, 2005. ACM Press.

[7] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen. The peer sampling service: experimental evaluation of unstructured gossip-based implementations. In *Middleware '04: Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, pages 79–98, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[8] M. Jelasity, A. Montresor, and O. Babaoglu. The bootstrapping service. In *ICDCSW '06: Proceedings of the 26th IEEE International ConferenceWorkshops on Distributed Computing Systems*, page 11, Washington, DC, USA, 2006. IEEE Computer Society.

[9] A. Montresor, M. Jelasity, and O. Babaoglu. Chord on demand. In *P2P '05: Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05)*, pages 87–94, Washington, DC, USA, 2005. IEEE Computer Society.