This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE P2P 2010 proceedings.

BitMON: A Tool for Automated Monitoring of the BitTorrent DHT

Konrad Jünemann, Philipp Andelfinger, Jochen Dinger, Hannes Hartenstein
Institute of Telematics & Steinbuch Centre for Computing
Karlsruhe Institute of Technology, Germany
Email: [juenemann, dinger, hartenstein]@kit.edu, philipp.andelfinger@student.kit.edu

***Abstract*—The distributed hash table (DHT) formed by BitTorrent has become very popular as a basis for various kinds of services. Services based on this DHT often assume certain characteristics of the DHT. For instance, for realizing a decentralized bootstrapping service a minimum number of peers running on a certain port are required. However, key characteristics change over time. Our measurements show that e.g. the number of concurrent users grew from 5 to over 7 millions of users during the last months. For making reliable assumptions it is thus essential to monitor the P2P network.**

This demo presents *BitMON*, a Java-based out-of-the-box platform for monitoring the BitTorrent DHT. This tool does not only crawl the network, but also automatically analyzes the collected data and visualizes the results. BitMON monitors the DHT's size in peers as well as the peers' IP addresses, port numbers, countries of origin and session length. Also, the long-term evolution of these indicators can be graphically displayed. Furthermore, BitMON is designed as a framework and can easily be extended or adapted to monitor other P2P networks.

I. Introduction

Peer-to-peer (P2P) networks like the Kademlia-based [6] distributed hash table (DHT) introduced by the BitTorrent Mainline Implementation have the ability to adapt to a changing number of participants (*churn*). Many use cases depend on specific properties of such networks that change over time and are thus hard to predict. For instance, for decentralized bootstrapping [2] a minimum number of peers running on a certain port are required. For companies such as BitTorrent, Inc. [1], it might be interesting what ratio of peers originated from a certain country. Therefore, monitoring of P2P networks has become increasingly important.

Several publications use P2P monitoring tools, e.g. [4], [8], [7]. While these publications focus on monitoring results, we try to ease scientific studies by automation of monitoring. This demo presents *BitMON*, an integrated out-of-the box platform for monitoring the BitTorrent DHT network. BitMON is designed to continuously crawl the network, automatically analyze the collected data and visualize the results.

II. BitMON – Overview

BitMON was developed as part of a project to monitor the BitTorrent DHT fully automatically over a longer period of time. The tool should allow operation by non-experts, but still collect and save all data necessary for scientific research.

The tool's main functionalities are provided by *modules*. Three types of modules exist:

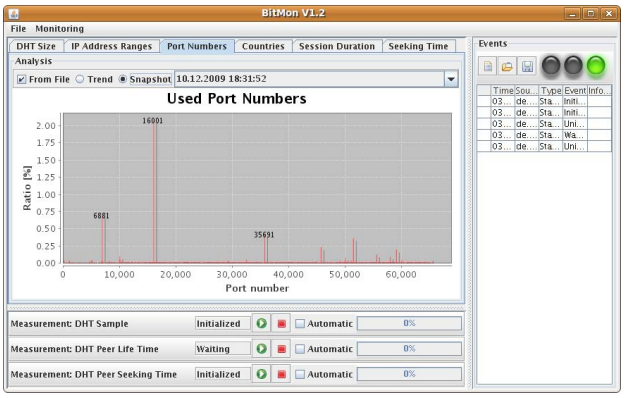

Fig. 1. Graphical user interface of BitMON

1) **MeasurementProviders** collect data about the P2P network, e.g. by crawling the DHT (see section III).
2) **Analyzers** analyze the collected data according to specific metrics and create plots as well as textual output.
3) **Thresholds** are used to check the analyzers' results for drastical changes. If a threshold was exceeded, a warning will be generated.

BitMON's graphical user interface (GUI) is shown in Fig. 1. On the left, the plots generated by the analyzers can be viewed in different tabs. Below, every MeasurementProvider offers a control panel. Different MeasurementProviders can be run simultaneously. On the right, a log is provided that displays events like exceeded thresholds or completed crawls.

Neither the data collection nor the analysis does require any human interaction: MeasurementProviders will be restarted automatically after they finish, analyzers will create plots as soon as data is collected and thresholds are checked automatically. Collected data and produced plots will be saved to disk for later inspection.

III. MeasurementProviders and Analyzers

BitMON includes three MeasurementProviders: The first one crawls a partition of the DHT to collect a representative sample of peers. The partition is crawled in multiple passes of increasing precision, until the number of found peers per pass falls below a certain threshold. The second MeasurementProvider links successively collected DHT samples to detect the starting times of the peers' sessions. These peers are

978-1-4244-7141-6/10/$26.00 ©2010 IEEE

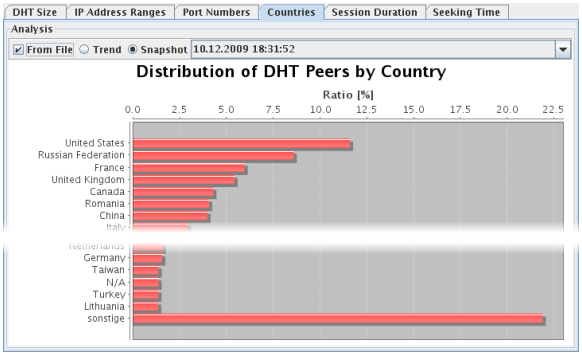

Fig. 2. Exemplary plot of the country snapshot analyzer

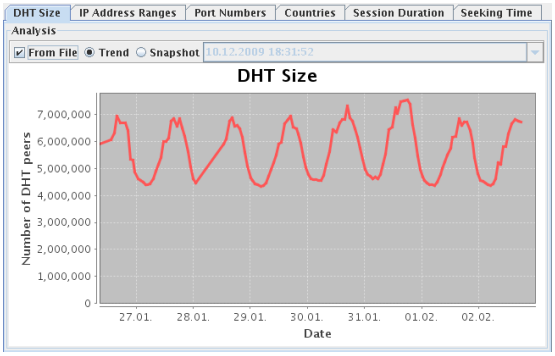

Fig. 3. Exemplary plot of the peer number trend analyzer

then regularly pinged to determine the ending times of their sessions. The third MeasurementProvider scans an IP range to collect data needed for the *Seeking time* analyzer.

BitMON currently comes with 12 analyzers, which can be divided into 6 *SnapshotAnalyzers* and 6 *TrendAnalyzers*. SnapshotAnalyzers process the results of a single run of a MeasurementProvider. Therefore they analyze momentary snapshots of the P2P network. TrendAnalyzers aggregate the results of several runs of a MeasurementProvider to a long-term trend. While each SnapshotProvider produces a row of independent plots, a TrendAnalyzer continually updates a single plot. In the following we briefly introduce the 6 Snapshot- and 6 TrendAnalyzers, currently offered by BitMON.

DHT size: Gives an estimation of the total number of peers the DHT consists of. The estimation is based on the number of peers found within a partition of the DHT. The trend illustrates the long-term evolution of this number (see Fig. 3).

IP address ranges: Analyzes the distribution of all found peers over different ranges of the IPv4 address space. For the trend, the address space is separated into partitions. Then, the change of the peers' distribution over these parts is plotted.

Port numbers: Depicts the distribution of all found peers over all valid port numbers. The trend diagram shows, how the ratio of peers running on the DHT's default port (e.g. port 6881) changes over time.

Country of origin: Illustrates the distribution of the found peers to their countries of origin as a bar chart (see Fig. 2). For the trend, the evolution of the ratio of the 6 most common countries will be shown.

Session duration: Plots what ratio of peers could be reached continuously for a certain time span. The trend shows the evolution of the peers' average length of a session.

Seeking time: Depicts the probability to find at least one peer after a certain time by scanning an IP range on the default port, i.e., port 6881 (see also [2]). The corresponding trend shows the development of this metric over time.

IV. Selected Features

Diagrams produced by BitMON are not static but can be interactively modified: In particular, any interesting area can be viewed in more detail, revealing even minor noticeable effects. At any time, diagrams as well as the data they are created from can be exported for external use. This way, diagrams can easily be replot with gnuplot or be made publicly available via a web interface. As the export can be performed automatically, it is even possible to create a website that continuously shows the DHT's evolution. BitMON can also be started without a GUI. Plots can still be automatically exported.

For good extensibility, BitMON was designed as a framework. In particular it is easy to add new analyzers and MeasurementProviders, enabling adaptation of different P2P networks. BitMON is written in Java and its source code is available on request. A modified version of the open source library *libtorrent* [5] is used for accessing BitTorrent. For mapping IPs to their countries of origin, the MaxMind GeoIP [3] library is used.

V. Demo

This demo is twofold. Firstly, we demonstrate live, how BitMON crawls BitTorrent and automatically analyzes and visualizes the results. Depending on the venue's technical infrastructure, BitMON will either be run locally on a machine at the conference's location or remotely at our university. We will present how the different MeasurementProviders and analyzers work in detail. Secondly, previously captured data will be presented. Among others, our measurements show that the number of concurrent peers grew from 5 to over 7 million during the last months. Furthermore, the ratio of peers running on the DHT's default port 6881 has dropped below 1% lately.

References

[1] BitTorrent Website. http://www.bittorrent.com/, 2010.
[2] J. Dinger and O. P. Waldhorst. Decentralized bootstrapping of p2p systems: A practical view. In *NETWORKING 2009*, pages 703–715, 2009.
[3] GeoIP Website. http://www.maxmind.com/app/ip-location, 2010.
[4] R. Jimenez, F. Osmani, and B. Knutsson. Connectivity properties of Mainline BitTorrent DHT nodes. In *IEEE Ninth International Conference on Peer-to-Peer Computing, (P2P '09)*, pages 262–270, 2009.
[5] libtorrent Website. http://www.rasterbar.com/products/libtorrent/, 2010.
[6] P. Maymounkov and D. Mazires. Kademlia: A peer-to-peer information system based on the xor metric. In *Peer-to-Peer Sys.*, pages 53–65, 2002.
[7] M. Steiner, T. En-Najjary, and E. W. Biersack. A global view of Kad. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 117–122, 2007.
[8] J. Yu, C. Fang, J. Xu, E.-C. Chang, and Z. Li. ID repetition in Kad. In *IEEE Ninth International Conference on Peer-to-Peer Computing, (P2P '09)*, pages 111–120, 2009.