Querying at Internet Scale

Brent Chun[‡], Joseph M. Hellerstein[†‡], Ryan Huebsch[†], Shawn R. Jeffery[†], Boon Thau Loo[†], Sam Mardanbeigi[†], Timothy Roscoe[‡], Sean Rhea[†], Scott Shenker[†§], and Ion Stoica[†]

[†]University of California at Berkeley, [‡]Intel Research Berkeley, and [§]International Computer Science Institute

p2p@db.cs.berkeley.edu

ABSTRACT

We are developing a distributed query processor called PIER, which is designed to run on the scale of the entire Internet. PIER utilizes a Distributed Hash Table (DHT) as its communication substrate in order to achieve scalability, reliability, decentralized control, and load balancing. PIER enhances DHTs with declarative and algebraic query interfaces, and underneath those interfaces implements multi-hop, in-network versions of joins, aggregation, recursion, and query/result dissemination. PIER is currently being used for diverse applications, including network monitoring, keyword-based filesharing search, and network topology mapping. We will demonstrate PIER's functionality by showing system monitoring queries running on PlanetLab, a testbed of over 300 machines distributed across the globe.

1. INTRODUCTION

The Internet is the largest distributed system ever built. The data available on the Internet includes not only information stored in centralized servers, but also at the "edges" of the network. Data resides on end-users' machines and in the control and monitoring systems that observe or collect the packets flowing on the network itself. This Internet data is updated in real time, while the set of data sources is constantly changing.

We are exploring the design of a decentralized, Internet-scale query processor called PIER. PIER (which stands for "Peer-to-Peer Information Exchange and Retrieval") is based on a confluence of database and network system design principles. It incorporates network technologies such as *Distributed Hash Tables* and *soft state*, along with adaptations of traditional distributed and parallel database execution technologies. We aim for PIER to scale to the realities of the full Internet, including geographic scalibility, dynamic memberhsip, as well as vast numbers of concurrent participants. Scaling to this degree has never previously been a goal of database research.

In this demonstration we show PIER operating on the PlanetLab wide-area testbed. PlanetLab currently consists of 300 machines worldwide. Though smaller than our design point in terms of the raw number of nodes, it serves as a concrete environment for demonstrating PIER running across the globe, and the largest, most realistic distributed platform available to researchers today. We will show a number of queries running on PIER that are used by PlanetLab researchers for monitoring the system on a regular basis.

2. OVERVIEW

PIER is a generic dataflow engine which has been outfitted with a set of relational query processing operators. This includes multi-hop, in-network algorithms for join, aggregation, and query/result dissemination. PIER can support both traditional query trees and DAGs, as well as cyclic graphs representing recursive queries. In addition to a "boxes and arrows" dataflow interface, PIER also provides a simple SQL interface, and support for continuous query variants of SQL.

One of the core technologies underlying PIER is a Distributed Hash Table (DHT). The term "DHT" is a catch-all for a set of schemes (e.g. [5, 7, 6]) sharing certain design goals. As the name implies, a DHT provides a hash table abstraction over multiple distributed compute nodes. Each node in a DHT can store data items, and each data item is identified by a key. At the heart of a DHT is an overlay routing scheme that delivers requests for a given key to a node currently responsible for that key. This is done without any global knowledge or permanent assignment of the mappings of keys to machines. Routing proceeds in a multi-hop fashion; each node maintains only a small set of neighbors, and routes messages to the neighbor that is in some sense "nearest" to the correct destination. The DHT automatically adjusts the mapping of keys and routing when the set of nodes changes.

The DHT forms the basis for communication in PIER. PIER also stores temporary tuples generated during query processing in the DHT. This provides PIER with a scalable and robust messaging substrate even when the set of nodes is dynamic.

PIER is currently being used for diverse applications, including network monitoring (the focus of this demonstration), as well as keyword-based filesharing search [3], and network topology analysis and routing using recursive queries [2]. A more complete description of PIER's design goals can be found in [1].

SIGMOD 2004 June 13-18, 2004, Paris, France.
Copyright 2004 ACM 1-58113-859-8/04/06 . . . $5.00.

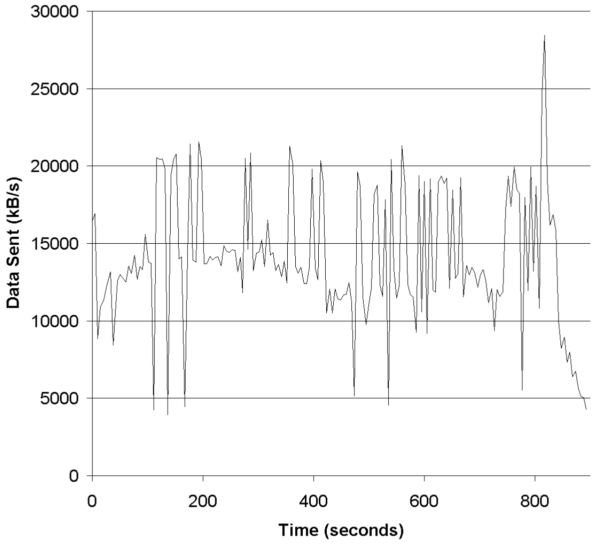

Figure 1: Continuous sum of outbound data rates over responding nodes running PIER on PlanetLab

Rule	Rule Description	Hits
1322	BAD-TRAFFIC bad frag bits	465,770
2189	BAD TRAFFIC IP Proto 103 (PIM)	123,558
1923	RPC portmap proxy attempt UDP	31,491
1444	TFTP Get	21,944
1917	SCAN UPnP service discover attempt	17,565
1384	MISC UPnP malformed advertisement	14,052
1321	BAD-TRAFFIC 0 ttl	10,115
1852	WEB-MISC robots.txt access	10,094
1411	SNMP public access udp	7,778
895	WEB-CGI redirect access	7,277

Table 1: The network-wide top ten intrusion detection rules reported by open-source Snort intrusion detection tools running locally at each node.

3. DEMONSTRATION

We are utilizing PlanetLab [4] for the first stage of PIER's deployment. PlanetLab is a consortium of over 100 institutions on five continents, enabling testing of distributed system designs in the context of all the real-world issues that arise in the global Internet. We view PlanetLab both as an infrastructure for testing PIER with realistic data and system parameters, and as a vehicle for "infecting" Internet research with database-inspired technologies.

Our demonstration will center on practical uses of PIER on PlanetLab, with a focus on queries used by other PlanetLab researchers to determine the state of the system. Two simple examples of such queries include suming the network traffic data rates over the nodes responding in the system (Figure 1), and the top ten network intrusions detected within the network (Table 1). Other PlanetLab monitoring queries include resource discovery services, system load monitoring, and distributed anomaly detection.

4. REFERENCES

[1] R. Huebsch, J. M. Hellerstein, N. L. Boon, T. Loo, S. Shenker, and I. Stoica. Querying the internet with pier. In *Proc. of VLDB 2003*, Sept. 2003.

[2] B. T. Loo, R. Huebsch, J. M. Hellerstein, T. Roscoe, and I. Stoica. Analyzing p2p overlays with recursive queries. Technical Report UCB/CSD-04-1301, UC Berkeley, Jan. 2004.

[3] B. T. Loo, R. Huebsch, I. Stoica, and J. Hellerstein. The Case for a Hyrid P2P Search Infrastructure. In *3rd International Workshop on Peer-to-Peer Systems (IPTPS'04)*, February 2004.

[4] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A blueprint for introducing disruptive technology into the Internet. In *Proc. ACM HotNets-I Workshop*, Princeton, Oct. 2002.

[5] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In *Proc. 2001 ACM SIGCOM Conference*, Berkeley, CA, August 2001.

[6] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz. Handling churn in a DHT. In *Proceedings of the USENIX Annual Technical Conference*, June 2004.

[7] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: Scalable Peer-To-Peer lookup service for internet applications. In *Proc. 2001 ACM SIGCOMM Conference*, pages 149–160, 2001.