Performance of Pastry in a Heterogeneous System

Fredrik Bjurefors   Lars-Åke Larzon   Richard Gold
Department of Information Technology
Uppsala University
{frbj3671,lln,rmg}@it.uu.se

Abstract

In this paper, we study how Pastry performs in a heterogeneous network environment of varying size. The large traffic overhead for management traffic makes the overlay nonfunctional if it grows too large. This can be circumvented by partitioning the routing tables at the cost of increased path lengths and response times.

1. Introduction

The motivation for this work is to study the performance of a proposed overlay architecture in a heterogeneous network environment. We have chosen to study Pastry, which is mainly motivated by the presence of a working simulator that could be easily extended with support for heterogeneous simulation scenarios. As Pastry share several design concepts with other proposed overlay architectures, we expect that our results can be generalized in a broader context.

2. Pastry

Pastry is a self-organized overlay that acts as a routing substrate[2]. In a network consisting of N nodes, Pastry can route to the numerically closest node to a given key in less than $\lceil \log_{2^b} N \rceil$ steps on average[1].

For the purpose of routing each Pastry node maintains a routing table and a leaf set. Each entry in the routing table contains the IP address of potentially many nodes whose *nodeId* have the appropriate prefix. A routing table entry is left empty if no node with a suitable *nodeId* prefix is known.

At each routing step, the node first checks to see if the key is within range of *nodeId*:s covered by its leaf set. If so, the message is directly forwarded to the destination. Otherwise, the node seeks to forward the message to the node with a longer matching prefix for the *nodeID*. If no such node exists, the message is forwarded to a node that is numerically closer to the key. If no suitable node exist in the leaf set, then the present node is the final destination for the message.

3. Simulations

3.1. Setup

Each node is configured with individual bandwidth, link latency and processing power. Simulation of bandwidth is done by calculating the time it takes a message to pass through a link plus the time between a message arrives at a node and when the link is available for transmission. Processing performance simulation is based on how many messages a node can process per millisecond.

Strong nodes in our simulations represent desktop computers connected through a 100 Mbit/s broadband link. Link latency is assumed to be 1 ms and no restrictions are made in terms of processing power. A *weak node* corresponds to a GPRS cellular phone. We assume a GPRS device classified as ’3+1’ [1], meaning that they can simultaneously listen to 3 downlink channels and transmit on 1 uplink channel. This is equivalent to a downlink bandwidth of 5 KB/s and uplink of 1.5 KB/s. Link latency is varying randomly from 100 to 500 milliseconds. The processing power of a weak node is also reduced to a capacity of 100 messages/second.

3.2. Simulation Methodology

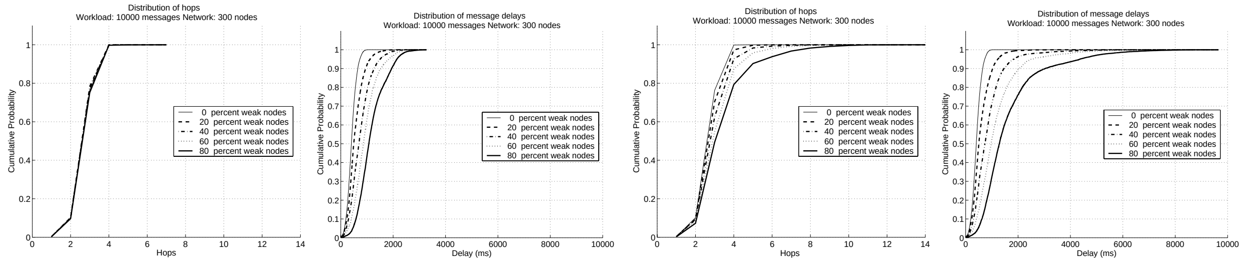
The simulator used in this study is the Microsoft Research Pastry Simulator v3.0A used on a Transit-stub topology with 600 routers, each attached to a LAN. For all network sizes, the Pastry nodes are randomly assigned to one of these LANs. The workload, used to create pastry nodes, in all our simulations are based on the gnutella workload [3].

We first measured the intensity of control messages during the initiation of all nodes for 10 different networks ranging in size from 30 to 2000 nodes. The purpose of this measurement is to get an idea of the management traffic load to start up an overlay network based on Pastry.

In our second set of measurements, we simulated startup and operation of networks with 30, 300 and 3000 nodes respectively with the amount of weak nodes ranging from 0 to 80% in steps of 20%. All these scenarios where then simulated with workloads ranging from 1000 to 100000 lookups during a 10-minute period which is the epoch of one simulation run.

The third set of measurements was essential identical to the second set, but with partitioned routing tables. An observation from the initial simulations was that all simula-

[1] b is a configuration parameter with a typical value of 4



(a) Full routing tables

(b) Partitioned routing tables for weak nodes

Figure 1. CDF plots of average path length and delay for a network of 300 nodes

tions could not be initiated due to a too high load on weak nodes in large overlays.

3.3. Results

Not all simulations could be initiated during our first simulation set. A closer look reveals increases in management traffic that cause weak nodes to congest. The major part of that traffic is produced through distance probes that measure the distance in delay to all nodes in the routing table. The network size when this problem occurs depends on the ratio of weak nodes.

When simulating the operation of a Pastry network, we measured the average path length and delay for all requests. CDF plots of these for a network size of 300 nodes and a workload of 10000 messages during each simulation run are shown in figure 1(a).

From the CDF plots, we see that the average path length increases by half a hop when going from 0% to 80% of weak nodes in the simulated overlay. However, the delay becomes roughly 3 times longer, which indicates that queues are being filled in intermediate nodes, most likely the weak ones.

To reduce the management traffic in the network, the routing tables in weak nodes are partitioned such that they contain approximately $\lceil log_{2^b} N \rceil \times (2^b - 1)/2$ entries. With this setup, we repeated the simulations and once again measured the average path lengths and delays. The results from these measurements are presented in figure 1(b).

By partitioning the routing table we reduce the management traffic in terms of distance probes due to less peers in the routing table. In exchange of reduced management traffic, message delays grows due to an increasing hop length. The increased hop length is a result of the partitioned routing table.

When we compare the message delay in homogenous network, with only strong nodes, to a network with 80 percent weak nodes (networks with 300 nodes), we can observe that the increase in average message delay is 800 milliseconds per message. The same comparison is made with the results from simulations with partitioned routing table(only in weak nodes) and here we can see an increase in delay with 1200 milliseconds per message. Comparing these two results, we can see that with the partitioned routing table the message delay increase with 35 percent on average per message.

In our largest simulations, with 3000 nodes in the network and a workload of 100,000 messages we observed significant performance degradations when partitioning the routing tables. Although the average hop length increased by only 0.4 hops when going from 0% to 80% weak nodes, the actual delay changed from an average of 450 ms to 1400 ms. Ten percent of the nodes experienced an increase from 700 ms on average to 2.7 seconds when increasing the average hop length with 1.2 hops. The highest measured delays are over 20 seconds. These increases in delay despite a modest increase in path length suggests that queues are building up with management traffic in weak nodes. However, the trick with partitioning the routing tables allows us to increase the overlay network with thousands of nodes before achieving the same congestion as happened for 2000 nodes when full routing tables were used.

4. Discussion

To make a routing substrate like Pastry work in heterogeneous scenarios, one must avoid creating bottlenecks in the routing system. In our case, this is the management traffic overhead in the overlay. By partitioning the routing tables, the management traffic is reduced but not eliminated. This essentially means that we reduce the problem by supporting construction of larger overlays before the management traffic will be overwhelming.

A more stable solution, however, would be to address heterogeneity in the design of overlay and peer-to-peer networks as they are becoming increasingly popular and important for new services.

Future work will include introducing intermittent connections, massive join/leave operations and roaming users as well as designing distributed data structures that are less vulnerable to suboptimal conditions.

References

[1] J. Cartwright R. Chakravorty and I. Pratt. Practical experience with TCP over GPRS. In *http://www.cl.cam.ac.uk/users/rc277/globe02.pdf.*

[2] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218, 2001.

[3] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Computing and Networking 2002 (MMCN '02)*, San Jose, CA, USA, January 2002.