p4Hadoop: Cloud Computing on the Web

Introduction

Hadoop is a framework that enables large-scale parallel computation and data processing on shared-nothing clusters of machines [3]. P4P is an architecture that allows for effective, co-operative traffic between network applications and network service providers [1]. P4P was originally designed to allow a peer-to-peer (P2P) application to utilize network level information provided by P4P portals to choose peers that provide high end-to-end bandwidth, do not congest the network and utilize cheaper links. Our project will integrate P4P policies with the Hadoop task scheduler to allow large scale data processing on the Internet without overloading network link or nodes.

Several applications could benefit from the merge of these tools. Users over the Internet may join a Hadoop cluster to contribute their free computing power for the scientific purposes (e.g. the SETI@home project) [6]. Another target application is Google Trends [5]. Google Trends is an on-line tool that allows a user to view data on how often a particular search term is entered in comparison to the total search volume across parts of the world. Google Trends is an analytical tool that allows the user to view statistics and compare the frequency and popularity of various search terms and their trends across the web.

Background

P4P stands for Proactive network Provider Participation for P2P [1, 2]. It is an extension of the popular peer-to-peer (P2P) applications. P2P applications enable data sharing without the need for a centralized server by forming peering links between computers. P4P optimizes these connections, using policies that determine the quality and cost-effectiveness of the connection given the network(s) topology. The idea is that Internet Service Providers and P2P software use these optimized connection to save the ISPs significant costs.

Hadoop is an open-source implementation of Google's MapReduce framework [3, 4]. The power of MapReduce lies in its programming paradigm that enables expressing arbitrary computation over data but hides the complexity of distributing load over possibly large number of heterogeneous machines in the network. Developers need only follow the programming model consisting of map() and reduce() functions and be able to describe data as a set of key/value pairs. This approach is extremely flexible and many real world tasks were implemented using MapReduce including indexing the Web for Google Search.

Central Idea

Hadoop is designed to run on shared-nothing clusters. We plan to extend Hadoop to run on the Internet
instead of on well-controlled clusters.

Hadoop specifies a centralized master known as a job tracker that controls which nodes execute map or reduce tasks. The tracker schedules map tasks to run on nodes that contain data to be processed. This preference for locality reduces computation time and network overhead associated with data transfer. If the data containing node is overloaded, the master schedules other nodes to execute the task. This phase is similar to P2P applications as the selected node establishes peer links with nodes that contain the data, retrieves the data and finally executes the map task.

P4P could provide the following network information to the job tracker to enable it to better schedule map, reduce tasks:

- For each overloaded data node that cannot run a map task, P4P will provide an ordered list of nodes (ordered by network metrics such as distance from data node, end-to-end bandwidth from data node and link costs) that could the execute map/reduce task in the absence of data locality.
- For each computation node that does not contain the data, P4P will provide an ordered list of peers (ordered by network metrics such as distance from data nodes, end-to-end bandwidth from data nodes and link costs) to download data from before executing the map/reduce task.
- High level policy and resource cost information. Clusters (such as Amazon EC2, Rackspace), network providers, individual nodes could provide computation, data transfer or data storage services at different rates. P4P could provide this information and the task scheduler could optimize task execution not only for performance but also for reduced costs.

**Project plan and resources**

1. Familiarize ourselves with the P4P architecture and P4P and Hadoop applications
2. Integrate P4P information into the Hadoop task scheduler
3. Compare the scheduling effectiveness of p4Hadoop with Hadoop on a networked environment based on several scheduling criteria
4. Document the results and write the project report

Our project will require the following resources. Note that this is a preliminary estimate of what is needed and we expect these requirements to change as the project develops.

1. A networked environment for testing the effectiveness of the scheduler in different network configurations. We require a minimum of two mobile nodes (including low capacity resources such as androids) and two fixed nodes on wired and wireless networks. The nodes need to run Unix operating systems and have a Java runtime environment.
2. Tools for measuring congestion and network performance
3. Access to computing clusters
4. Access to configured P4P networks

**Project Deliverables**

i. New Hadoop scheduler
ii. Any modified Hadoop and/or P4P source code
iii. Project report, including:
   - summary of our findings
   - test performance results with analysis
   - a recommendation for how our solution can be used to improve large scale data processing on the Internet

References


