Grant Proposal

Andrew Rosen

Abstract

Distributed Hash Tables (DHTs) are protocols and frameworks used by peer-to-peer (P2P) systems. They are used as the organizational backbone for many P2P file-sharing systems due to their scalability, fault-tolerance, and load-balancing properties. These same properties are highly desirable in a distributed computing environment, especially one that wants to use heterogeneous components.

We show that DHTs can be used not only as the framework to build a P2P file-sharing service, but as a P2P distributed computing platform. We propose creating a P2P distributed computing framework using distributed hash tables, based on our prototype system ChordReduce. This framework would make it simple and efficient for developers to create their own distributed computing applications. Unlike Hadoop and similar MapReduce frameworks, our framework can be used both in both the context of a datacenter or as part of a P2P computing platform. This opens up new possibilities for building platforms to distributed computing problems.

One advantage our system will have is an autonomous load-balancing mechanism. Nodes will be able to independently acquire work from other nodes in the network, rather than sitting idle. More powerful nodes in the network will be able use the mechanism to acquire more work, exploiting the heterogeneity of the network.

We plan on running large scale experiments using both cloud-computing platforms and small, consumer-grade hardware. This variety of computing device will establish the flexibility of our framework and its ability to use the heterogeniety of the network to its advantage.

By utilizing the load-balancing algorithm, a datacenter could easily leverage additional P2P resources at runtime on an as needed basis. Our framework will allow MapReduce-like or distributed machine learning platforms to be easily deployed in a greater variety of contexts.

1 What are DHTs

My dissertation focusing on creating a generalized framework for distributed computing based on Distributed Hash Tables (DHTs). Distributed Hash Tables are a tool which allows a computers (nodes) to self-organize into a decentralized network for storing and retrieving data.

There are many different types of DHTs, but all DHTs share very important qualities. They are *scalable*, which means that each additional node in the network minimally impacts the cost of keeping the network organized. DHTs are also highly *fault-tolerant*. Unlike many other systems, DHTs assume that nodes will be continuously entering and leaving the network. Because of the way DHTs are organized, DHTs can handle large scale failures, such as power outage affecting an entire city. The last quality of DHTs is that they are *load-balancing*. This mean the data the is stored in a DHTs is evenly distributed among nodes in the network.

2 What Have We done so far and why should the reader care

All of these qualities are highly desirable in a distributed computing, so it was a natural step for us to build distributed computing framework based on DHTs. We have shown in previous work that DHTs can be used to distribute work among nodes the same way data is distributed [1].

3 What do we plan on doing

3.1 Why is this exciting

Some example problems are Monte-Carlo Methods (of interest to mathematicians and economists), Our framework would not only be of interest to computer scientists. The "plug-and-play" nature we envision would make it easy for experts in other fields to use our framework to solve computationally intensive problems.

4 What do I need funding for?

Heterogeneous networks

We have successfully developed our software to achieve this, but need funding to do real world network tests. These tests would allow us to experimentally show that our software conclusively works on a large scale and on heterogeneous networks.

I need hardware of different types and locations.

This project would help by enabling allowing both organizations with large amounts of computing power and average developers with a fewer resources spend less time setting up and configuring their hardware to work together; the goal is for our software to make distributed computing more of matter of "plug-and-play."

References

[1] Andrew Rosen, Brendan Benshoof, Robert W Harrison, and Anu G. Bourgeois. Mapreduce on a chord distributed hash table. In *2nd International IBM Cloud Academy Conference*.

5 Vita goes here

Education

Ph.D. in Computer Science, Georgia State University. May 2016 (Expected)
M.S. in Computer Science, Georgia State University. May 2014, 3.89 GPA
B.S. in Computer Science, Georgia Institute of Technology. May 2010, 3.00 GPA
Minor in Music, Georgia Institute of Technology. May 2010

Appointments

2CI Astroinformatics Fellow, Georgia State University, Aug 2012 - Present
Graduate Research Assistant, Georgia State University, Aug 2011 - Present
Graduate Lab Assistant, Georgia State University, May 2011 - 2013

Publications

1. Andrew Rosen, Brendan Benshoof, Robert W. Harrison, Anu G. Bourgeois "MapReduce on a Chord Distributed Hash Table" Presentation ICA CON 2014, Poster at IPDPS 2014 PhD Forum

2. Brendan Benshoof, Andrew Rosen, Anu G. Bourgeois, Robert W. Harrison "VHASH: Spatial DHT based on Voronoi Tessellation" ICA CON 2014

3. Erin-Elizabeth A. Durham, Andrew Rosen, Robert W. Harrison "A Model Architecture for Big Data applications using Relational Databases" 2014 IEEE BigData - C4BD2014 - Workshop on Complexity for Big Data

4. Chinua Umoja, J.T. Torrance, Erin-Elizabeth A. Durham, Andrew Rosen, Dr. Robert Harrison "A Novel Approach to Determine Docking Locations Using Fuzzy Logic and Shape Determination" 2014 IEEE BigData - Poster and Short Paper

5. Erin-Elizabeth A. Durham, Andrew Rosen, Robert W. Harrison "Optimization of Relational Database Usage Involving Big Data" IEEE SSCI 2014 - CIDM 2014 - The IEEE Symposium Series on Computational Intelligence and Data Mining

6. Brendan Benshoof, Andrew Rosen, Anu G. Bourgeois, Robert W. Harrison "A Distributed Greedy Heuristic for Computing Voronoi Tessellations With Applications Towards Peer-to-Peer Network" IEEE IPDPS 2015 - Workshop on Dependable Parallel, Distributed and Network-Centric Systems

7. Chaoyang Li, Andrew Rosen, and Anu G. Bourgeois "A Novel Approach to Efficiently Detect 3D Full-View Coverage for Camera Sensor Networks" Submitted to IPDPS 2016

8. Andrew Rosen, Brendan Benshoof, Robert W. Harrison, Anu G. Bourgeois "UrDHT: A Unified Model for Distributed Hash Tables" Submitted to IPDPS 2016

9. Andrew Rosen, Brendan Benshoof, Robert W. Harrison, Anu G. Bourgeois "The Sybil Attack on Peer-to-Peer Networks From the Attacker?s Perspective" In preparation

Service

Vice Chair, Georgia State University Chapter of the Association for Computing Machinery (ACM), May 2012 - Present
Treasurer, Georgia State University Chapter of the Association for Computing Machinery (ACM), May 2012 - May 2014
New Graduate Student Orientation Panelist, Georgia State University, 2014-2015
Department Representative, Georgia State University Arts and Sciences Tech Fee Committee, 2013 - 2015

External Funding

TCPP Travel Grant for IPDPS 2014

Honors and Awards

Outstanding Graduate Student Teaching Award, Georgia State University, 2014

Employment

Instructor, Georgia State University, Spring 2012 (CSc 3410), Spring 2013,
Developer, Georgia Tech Sonification Lab, Atlanta, GA May-Dec 2010
Undergraduate Researcher, Georgia Tech Sonification Lab, Atlanta, GA Fall 2007 -Dec 2009

6 Budget and Justification

Item	Amount requested
Cloud Computing Server Time	$1200
10x Raspberry PI 2	$400
Raspberry PI peripherals	$250
Total Amount requested	$2000

The base cost for a cloud computing machine ranges between $0.008 and $0.075 per machine per hour. We need at least 100 machines for at least 100 hours of experimentation at the bare minimum. Buying cloud computing time from both Google and Amazon would allow us to do realistic networks

However, we still need more machines in order to test heterogeneity in the network. Raspberry PIs allow us to test the lower end of computing power. Raspberry PIs are credit-card sized computers which are powerful enough to test our software, but much weaker in computational power compared to what we can purchase from Amazon. They would allow us to our autonomous load-balancing and see how well our algorithms distribute work between stronger and weaker nodes. Furthermore, we could attach some to different network interfaces, making some wired and some wireless, which would allow us to further vary our network topology.