Proposal Defense
Towards a Framework for DHT Distributed Computing

Andrew Rosen

Georgia State University

July 15th, 2015

Table of Contents

Objective

Our objective is to create a generalized framework for distributed computing using Distributed Hash Tables.

What is Distributed Computing

Challenges

- Scalability

Challenges

- Scalability
- Fault-Tolerance

Challenges

- Scalability
- Fault-Tolerance
- Load-Balancing

Distributed Key/Value Stores

Current Applications

Applications that use or incoperate DHTs:

- P2P File Sharing applications, such as Bittorrent [1] [4].
- Distributed File Storage [2].
- Distributed Machine Learning [3].
- Name resolution in a large distributed database [5].

Strengths of DHTs

Table of Contents

Functions

put(*key*, *value*) Stores *value* at the node responsible for *key*, where $key = hash(value)$.

get(*key*) Returns the *value* associated with *key*.

lookup(*key*) Finds the node responsible for a given key.

Chord

content...

Kademlia

VHash

content...

Table of Contents

ChordReduce

content

DGVH

content

Sybil Analysis

content

Table of Contents

UrDHT

This kind of framework does not exist.

DHT Distributed Computing

content

Autonomous Load-Balancing

content

Bram Cohen.
Incentives build robustness in bittorrent.
In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72, 2003.

Frank Dabek, M Frans Kaashoek, David Karger, Robert Morris, and Ion Stoica.
Wide-Area Cooperative Storage with CFS.
ACM SIGOPS Operating Systems Review, 35(5):202–215, 2001.

Mu Li, Li Zhou, Zichao Yang, Aaron Li, Fei Xia, David G Andersen, and Alexander Smola.
Parameter server for distributed machine learning.

Andrew Loewenstern and Arvid Norberg.
BEP 5: DHT Protocol.
http://www.bittorrent.org/beps/bep_0005.html, March 2013.

Gabriel Mateescu, Wolfgang Gentzsch, and Calvin J. Ribbens.
Hybrid computing—where {HPC} meets grid and cloud computing.

Future Generation Computer Systems, 27(5):440 – 453, 2011.