Introduction
000000000

Background
0000

Previous Work
000

Proposed Work
000

Proposal Defense
Towards a Framework for DHT Distributed Computing

Andrew Rosen

Georgia State University

July 15th, 2015

Georgia State University

Introduction
○○○○○○○○○

Background
○○○○

Previous Work
○○○

Proposed Work
○○○

Table of Contents

Georgia State
University

Objective

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

Georgia State
University

Objective

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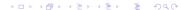Distributed Hash Tables.

Or

Georgia State
University

Objective

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

Or

We want to build a completely decentralized distributed computing framework.

Georgia State
University

What do I Mean by Distributed Computing?

A system where we can take a task and break it down into multiple parts, where each part is worked upon individually.

Georgia State
University

Challenges of Distributed Computing

Distributed Computing platforms should be:

Scalable The larger the network, the more resources need to be spent on maintaining and organizing the network.

Remember, computers aren't telepathic. There's always an overhead cost. It will grow. The challenge of scalability is designing a protocol that grows this organizational cost at an extremely slow rate. For example, a single node keeping track of all members of the system might be a tenable situation up to a certain point, but eventually, the cost becomes too high for a single node.

Challenges of Distributed Computing

Distributed Computing platforms should be:

Scalable The larger the network, the more resources need to be spent on maintaining and organizing the network.

Fault-Tolerant As we add more machines, we need to be able to handle the increased risk of hardware failure.

Georgia State University

Challenges of Distributed Computing

Distributed Computing platforms should be:

Scalable The larger the network, the more resources need to be spent on maintaining and organizing the network.

Fault-Tolerant As we add more machines, we need to be able to handle the increased risk of hardware failure.

Hardware failure is a thing that can happen. Individually the chances are low, but this becomes high when we're talking about millions of machines. Also, what happens in a P2P environment.

Distributed Computing and Challenges

Challenges of Distributed Computing

Distributed Computing platforms should be:

Scalable The larger the network, the more resources need to be spent on maintaining and organizing the network.

Fault-Tolerant As we add more machines, we need to be able to handle the increased risk of hardware failure.

Load-Balancing Tasks need to be evenly distributed among all the workers.

Georgia State
University

Challenges of Distributed Computing

Distributed Computing platforms should be:

Scalable The larger the network, the more resources need to be spent on maintaining and organizing the network.

Fault-Tolerant As we add more machines, we need to be able to handle the increased risk of hardware failure.

Load-Balancing Tasks need to be evenly distributed among all the workers.

If we are splitting the task into multiple parts, we need some mechanism to ensure that each worker gets an even (or close enough) amount of work.

What Are Distributed Hash Tables?

Distributed Key/Value Stores

Distributed Hash Tables are mechanisms for storing values associated with certain keys.

- Values, such as filenames, data, or IP/port combinations are associated with keys.
- These keys are generated by taking the hash of the value.
- We can get the value for a certain key by asking any node in the network.

Georgia State
University

Current Applications

Applications that use or incorporate 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].

Georgia State University

How Does It Work?

We'll explain in greater detail later, but briefly:

- DHTs organize a set of nodes, each identified by an ID (their key).
- Nodes are responsible for the keys that are closest it their IDs.
- Nodes maintain a list of other peers in the network.
  - Typically a size $\log(n)$ subset of all nodes in the network.
- Each node uses a very simple routing algorithm to find a node responsible for any given key.

Georgia State
University

How Does It Work?

We'll explain in greater detail later, but briefly:

- DHTs organize a set of nodes, each identified by an ID (their key).
- Nodes are responsible for the keys that are closest it their IDs.
- Nodes maintain a list of other peers in the network.
  - Typically a size $\log(n)$ subset of all nodes in the network.
- Each node uses a very simple routing algorithm to find a node responsible for any given key.

We use ID for nodes and keys for data so we always know our context.

Introduction
ooooooo●oo

Background
oooo

Previous Work
ooo

Proposed Work
ooo

What Are Distributed Hash Tables?

Strengths of DHTs

DHTs are designed for large P2P applications, which means they need to be (and are):

Scalable

Fault-Tolerant

Load-Balancing

Georgia State University

- Remember to mention Napster.
- Distributed Hash Tables were designed to be used for completely decentralized P2P applications involving millions of nodes.
- As a result of the P2P focus, DHTs have the following qualities

Introduction
○○○○○○○●○

Background
○○○○

Previous Work
○○○

Proposed Work
○○○

Why DHTs and Distributed Computing

DHTs Address the Specified Challenges

Uses For DHT Distributed Computing

- Embarrassingly Parallel Computations
  - Brute force cryptography.
  - Genetic algorithms.
  - Markov chain Monte Carlo methods.
  - Any problem that can be framed using Map and Reduce.
- Can be used in either a P2P context or a more traditional deployment.

Georgia State
University

Table of Contents

Georgia State University

Introduction
00000000

Background
●000

Previous Work
000

Proposed Work
000

The Components and Terminology

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.

Georgia State
University

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*.

- There is usually a delete function as well, but it's not important.

Introduction
ooooooooo

Background
o●oo

Previous Work
ooo

Proposed Work
ooo

Example DHTs

Chord

content...

Georgia State
University

Introduction
○○○○○○○○○

Background
○○●○

Previous Work
○○○

Proposed Work
○○○

Example DHTs

Kademlia

Introduction
oooooooooo

Background
oooo●

Previous Work
ooo

Proposed Work
ooo

Example DHTs

VHash

Maybe, if room

Table of Contents

Georgia State University

ChordReduce

content

Introduction
000000000

Background
0000

Previous Work
0●0

Proposed Work
000

VHash

DGVH

content

Introduction
○○○○○○○○○

Background
○○○○

Previous Work
○○●

Proposed Work
○○○

Sybil

Sybil Analysis

content

Georgia State
University

Table of Contents

UrDHT

This kind of framework does not exist.

Georgia State University

DHT Distributed Computing

content

Introduction
○○○○○○○○○

Background
○○○○

Previous Work
○○○

Proposed Work
○○●

Autonomous Load-Balancing

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.