

DHT Distributed Computing

└ Introduction

└ Distributed Computing and Challenges

└ Challenges of Distributed Computing

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.

DHT Distributed Computing

└ Introduction

└ Distributed Computing and Challenges

└ Challenges of Distributed Computing

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.

DHT Distributed Computing

└ Introduction

└ Distributed Computing and Challenges

└ Challenges of Distributed Computing

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.

DHT Distributed Computing

└ Introduction

└ What Are Distributed Hash Tables?

└ How Does It Work?

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 to 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.

DHT Distributed Computing

└ Introduction

└ What Are Distributed Hash Tables?

└ Strengths of DHTs

Strengths of DHTs

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

Scalable
Fault-Tolerant
Load-Balancing

- 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



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.