Load balancing models for DHT-based Peer-to-Peer Networks

draft-harjula-p2psip-loadbalancing-survey-00

E.Harjula, M. Ylianttila

Motivation

- Load balancing is an essential functionality to provide fair load distribution between peer nodes
- As shown by myriad of research efforts, DHTs' consistent hashing is not sufficient for providing fair load balancing in dynamic heterogeneous networks

→ *Additional load balancing mechanism is needed*

- Previous activities concerning load balancing at the P2PSIP group, without thorough background work

→ Survey of existing and proposed load balancing mechanisms is needed

Overview

- Overall load balancing process
  1. Measure load
  2. Distribute load information
  3. Balance the load
- P2P load balancing models usually implement the phases 2 and 3 of the above process
- Tens of existing or proposed load balancing models
- Fundamental models
  1. Using virtual servers
  2. Controlling the object location
  3. Controlling the node location
  4. Balancing the address-space

Virtual servers

- Main idea: Multiple virtual servers (overlay node instances) per node
  - → Static balancing effect
- Varying number of VNs allocated per node
  - → Node capability -aware load balancing
- Dynamic VN reallocation/migration
  - → Reactive load balancing

Controlling object location

- During the insertion, object is placed on the least loaded of several candidate nodes (power of n choices)
  - → Static, capability-aware balancing effect
- Dynamic object relocation
  - → Reactive load balancing

Controlling node location

- During runtime, nodes compare their load with other nodes. When needed, lightly loaded nodes may relocate to split the address space of the heavier loaded nodes.
  - → Dynamic, reactive capability-aware balancing effect

Address-space balancing

- Each node has a fixed set of possible locations on the overlay, of which one providing the address space closest to the system average is selected
  - → Dynamically balances the address space among the nodes
- (Somewhat similar to controlling node location, but different goal)

Brief analysis

Fundamental LB Model	Achievable load balance quality	Responsiveness to rapid load changes	Node capability awareness	Cost
Virtual server	High	High if dynamic VN reallocation, otherwise no responsiveness at all	Yes if varying nr of VNs/node, otherwise no	▪**Very high maintenance overhead** ▪(Also transfer cost with dynamic VN reallocation)
Controlling object location	High	High if dynamic object relocation, otherwise low	Yes	▪**No additional maintenance overhead** ▪Object lookup overhead (lower lookup overhead & increased maintenance overhead if redirection pointers in use) ▪Multiple hash generation in object insertion & lookup ▪(Also transfer cost with dynamic object relocations)
Controlling node location	High	High, if heavy node probes, otherwise low	Yes	▪Maintenance overhead ▪Node lookup overhead ▪Transfer cost in node relocations
Address-space balancing	Moderate	No	No	▪Maintenance overhead ▪Node lookup overhead ▪Transfer cost in virtual node relocations

Discussion

- Comments/Questions?