

Autonomous Dynamic Replica Placement for Load Balancing

CS 848 Project Presentation

Chen Zhang

2006-11-29

Problem Definition

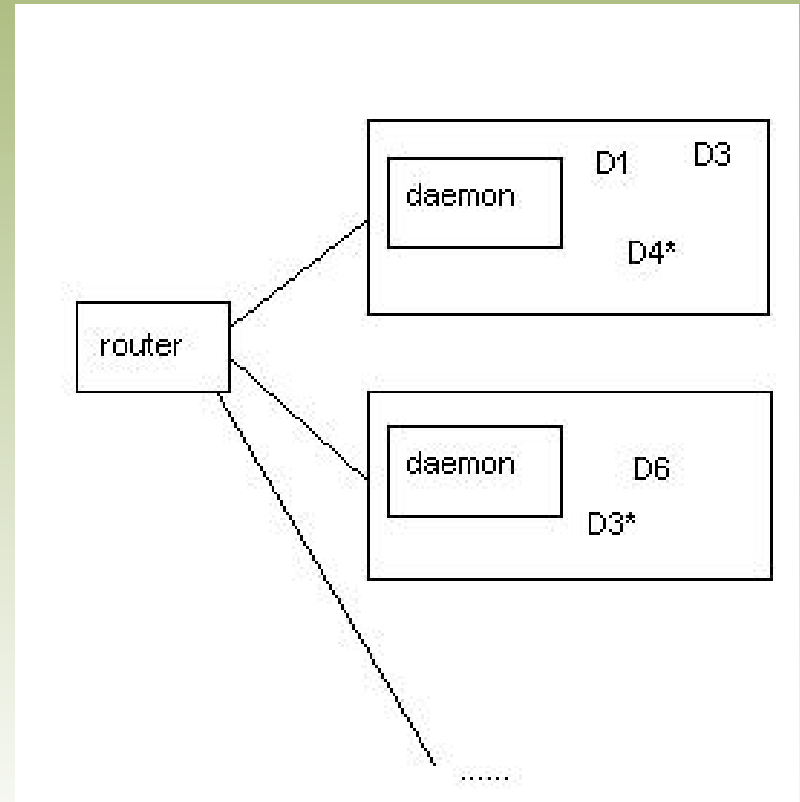
- Input:
 - $\{T: T1, T2, \dots, Tn\}$: a set of transaction types
 - $\{M: M1, M2, \dots, Mn\}$: a set of machines
 - $\{D: D1, D2, \dots, Dn\}$: a set of data replica
- Output:
 - A placement scheme mapping T, D to M
- Goal:
 - Load on M is balanced dynamically
 - No central control. M cooperates in finding balance.

Term Definition

- Resource Constraint R
 - Key-value pairs for hw/sw requirements
- Load L: used ratio for key elements.
 - $L(\text{cpu, mem, hard disc, bandwidth})$
- Transaction type t
 - $t(\text{request rate, throughput requirement, } D, R)$
 - $\{M: M1, M2, \dots, Mn\}$: a set of machines
 - $\{D: D1, D2, \dots, Dn\}$: a set of data replica
- Machine m
 - $m(L, R)$
- Data replica d
 - $d(\text{data, IsMaster})$

Solution

- Placement scheme S
 - $S(I, P)$
 - I: initial placement
 - P: dynamic placement
- System building blocks
 - A p2p overlay
 - A daemon on every M
 - Front-end request router



How it works

- When a new request r comes in
 - Router dispatched r to a random machine m
 - The daemon at m check if m is ok to serve r
 - If not, the daemon will start a search for suitable machine to serve r , pick a random one from the result set and forward r there
- Placement is adjusted dynamically by daemons

Note:
The overall placement is totally transparent

daemon

- Decentralized/local search
- Forward request elsewhere
- Load prediction
- Capable of transferring data between each other
- Deploying placement
 - When given I to deploy
 - When slashdot occurs

Feasibility and Novelty

- Feasibility
 - Use existing p2p overlay
 - Use existing decentralized search method
 - Use existing method for load prediction
 - Only need a daemon
- Novelty
 - Avoid solving NP-complete prob.
 - Avoid devising approximation algorithms
 - Fully transparent placement
 - Autonomous placement management

Problems

- Those intrinsic to the existing techs
 - Good thing is we decompose the problem into individual tasks such as load prediction, decentralized search, etc. that we can work on those individual topics respectively
- Slashdot?
 - Solution: a lazy and a bit wiser router
 - It holds request for a very small interval
 - If slashdot, then extract the mostly requested type of transaction and inform a random daemon first.