Efficiency of Tree-structured Peer-to-peer Service Discovery Systems

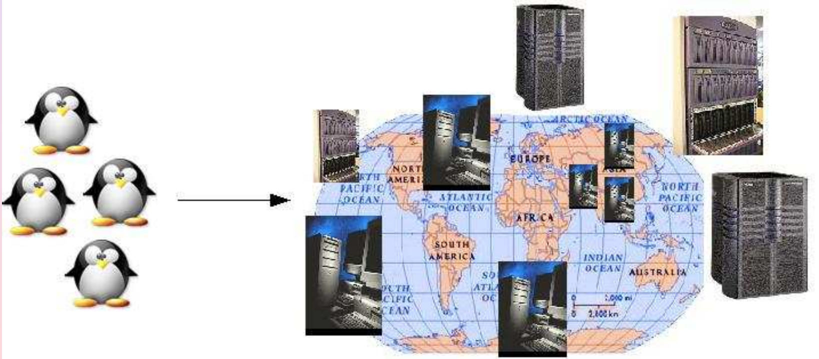
Cédric Tedeschi, Eddy Caron, Frédéric Desprez

University of Lyon, France
LIP laboratory. UMR CNRS-ENS Lyon-UCB Lyon-INRIA 5668

Hot-P2P, April 18, 2008

Initial context

- Service discovery in grid computing
  - Service (binary file, library) installed on servers
  - Servers declare their services, client discovers them
  - Need for maintaining this information



Initial context

- Service discovery in grid computing
  - Service (binary file, library) installed on servers
  - Servers declare their services, client discovers them
  - Need for maintaining this information
- Target platforms
  - large scale
  - no central infrastructure
  - dynamic joins and leaves of nodes
- P2P systems
  - Purely decentralized algorithms
  - Scalable algorithms to retrieve objects
  - Fault-tolerance

Trie-based overlays

A promising way to store and retrieve services

- Advantages
  - Efficient range queries
  - Automatic completion of partial strings
  - Easy extension to multi-attribute queries
- Existing approaches
  - Skip Graphs (Aspnes and Shah – 2003)
  - P-Grid (Datta, Hauswirth, John, Schmidt, Aberer – 2003)
  - PHT (Ramabhadran, Ratnasamy, Hellerstein, Shenker – 2004)
  - DLPT (Caron, Desprez, Tedeschi – 2005)
  - Nodewiz (Basu, Banerjee, Sharma, Lee – 2005)

Trie-based overlays

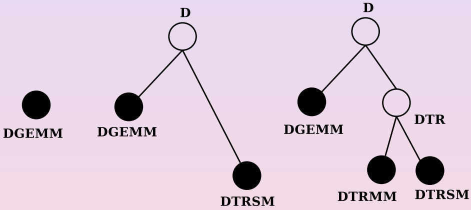
A promising way to store and retrieve services

- Advantages
  - Efficient range queries
  - Automatic completion of partial strings
  - Easy extension to multi-attribute queries
- Existing approaches
  - Skip Graphs (Aspnes and Shah – 2003)
  - P-Grid (Datta, Hauswirth, John, Schmidt, Aberer – 2003)
  - PHT (Ramabhadran, Ratnasamy, Hellerstein, Shenker – 2004)
  - DLPT (Caron, Desprez, Tedeschi – 2005)
  - Nodewiz (Basu, Banerjee, Sharma, Lee – 2005)

DLPT : A trie-based indexing system

Logical structure

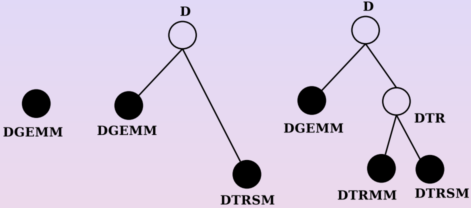
- Greatest Common Prefix Tree
- Dynamically constructed
- Bounded degree and height



DLPT : A trie-based indexing system

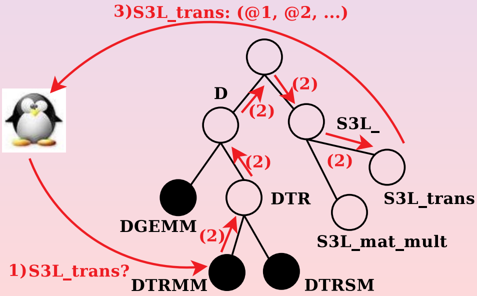
Logical structure

- Greatest Common Prefix Tree
- Dynamically constructed
- Bounded degree and height



Lookup

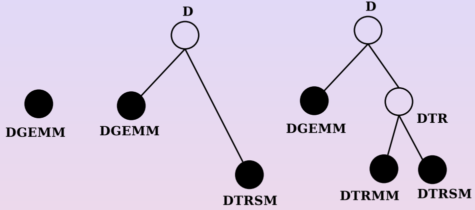
- Exact match
- Autocompetion
- Range queries



DLPT : A trie-based indexing system

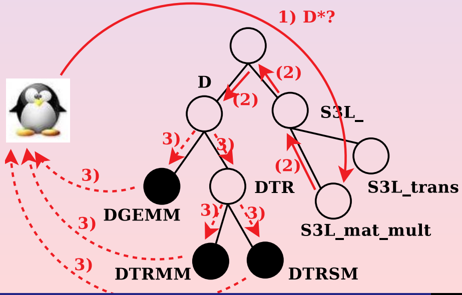
Logical structure

- Greatest Common Prefix Tree
- Dynamically constructed
- Bounded degree and height



Lookup

- Exact match
- Autocompetion
- Range queries



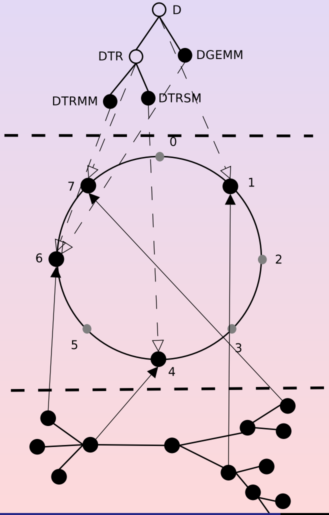
DLPT : Mapping of the system onto the network

Principles

- An underlying DHT
- Each physical node (*Peer*) runs one or more logical process

Drawbacks

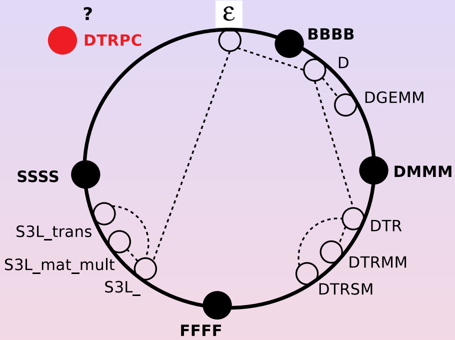
- Need for a DHT
  - Important global cost
  - Randomly achieved
- Load unbalance
  - Depth of nodes in the tree
  - Heterogeneous popularity of services, hot spots



Objectives

1. Avoid the need for a DHT to reduce the cost
2. Inject some load balancing

Build a ring over the physical network



- Build a Chord-like ring over peers to distribute the tree nodes
- Use the tree links to maintain the physical network
- Complexity : Trie complexities + 2

Peer insertion algorithm

1. Joining peer P contacts a random tree node
2. Route the request to the tree node N s.t. ID_N has the highest id lower than ID_P.
3. The succeqssor of P is either P_N or $succ(P_N)$

Load balancing heuristics - related work

- Karger and Ruhl, 2001
  - periodic random item balancing
  - homogeneity of peer capacities
- Godfrey *et al.*, 2003
  - periodic item redistribution
  - semi-centralized
- Ledlie and Seltzer, 2005
  - Based on the k-choices principle
  - heterogeneity of peer capacities and data popularity
  - chooses the best location for a joining peer among k

A novel heuristic : *Max Local Throughput*
Objective function

- At time unit τ
- Two adjacent peers S and P with capacity C_S and C_P
- ν_S^τ and ν_P^τ the sets of nodes currently managed by S and P.
- $L_S^\tau = \sum_{n \in \nu_S^\tau} l_n$
- $L_P^\tau = \sum_{n \in \nu_P^\tau} l_n$
- $T_{S,P}^\tau = min(L_S^\tau, C_S) + min(L_P^\tau, C_P)$
- We want to maximize the throughput of time unit $\tau+1$ based on knowledge of time unit τ
- Find $\nu_S^{\tau+1}$ and $\nu_P^{\tau+1}$ that maximizes $T_{S,P}^{\tau+1}$.

A novel heuristic : *Max Local Throughput*
Objective function

- At time unit τ
- Two adjacent peers S and P with capacity C_S and C_P
- ν_S^τ and ν_P^τ the sets of nodes currently managed by S and P.
- $L_S^\tau = \sum_{n \in \nu_S^\tau} l_n$
- $L_P^\tau = \sum_{n \in \nu_P^\tau} l_n$
- $T_{S,P}^\tau = min(L_S^\tau, C_S) + min(L_P^\tau, C_P)$
- We want to maximize the throughput of time unit $\tau + 1$ based on knowledge of time unit τ
- Find $\nu_S^{\tau+1}$ and $\nu_P^{\tau+1}$ that maximizes $T_{S,P}^{\tau+1}$.

A novel heuristic : *Max Local Throughput*
Objective function

- At time unit τ
- Two adjacent peers S and P with capacity C_S and C_P
- ν_S^τ and ν_P^τ the sets of nodes currently managed by S and P.
- $L_S^\tau = \sum_{n \in \nu_S^\tau} l_n$
- $L_P^\tau = \sum_{n \in \nu_P^\tau} l_n$
- $T_{S,P}^\tau = min(L_S^\tau, C_S) + min(L_P^\tau, C_P)$
- We want to maximize the throughput of time unit $\tau+1$ based on knowledge of time unit τ
- Find $\nu_S^{\tau+1}$ and $\nu_P^{\tau+1}$ that maximizes $T_{S,P}^{\tau+1}$.

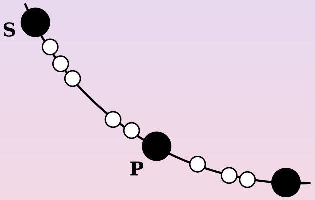
A novel heuristic : *Max Local Throughput*
Objective function

- At time unit τ
- Two adjacent peers S and P with capacity C_S and C_P
- ν_S^τ and ν_P^τ the sets of nodes currently managed by S and P.
- $L_S^\tau = \sum_{n \in \nu_S^\tau} l_n$
- $L_P^\tau = \sum_{n \in \nu_P^\tau} l_n$
- $T_{S,P}^\tau = min(L_S^\tau, C_S) + min(L_P^\tau, C_P)$
- We want to maximize the throughput of time unit $\tau+1$ based on knowledge of time unit τ
- Find $\nu_S^{\tau+1}$ and $\nu_P^{\tau+1}$ that maximizes $T_{S,P}^{\tau+1}$.

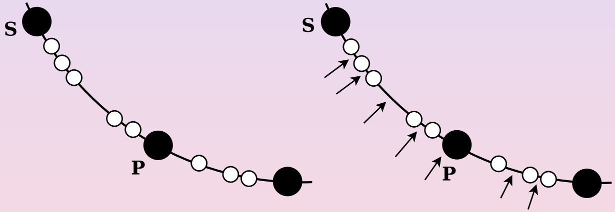
A novel heuristic : *Max Local Throughput*
Objective function

- At time unit τ
- Two adjacent peers S and P with capacity C_S and C_P
- ν_S^τ and ν_P^τ the sets of nodes currently managed by S and P.
- $L_S^\tau = \sum_{n \in \nu_S^\tau} l_n$
- $L_P^\tau = \sum_{n \in \nu_P^\tau} l_n$
- $T_{S,P}^\tau = min(L_S^\tau, C_S) + min(L_P^\tau, C_P)$
- We want to maximize the throughput of time unit $\tau+1$ based on knowledge of time unit τ
- Find $\nu_S^{\tau+1}$ and $\nu_P^{\tau+1}$ that maximizes $T_{S,P}^{\tau+1}$.

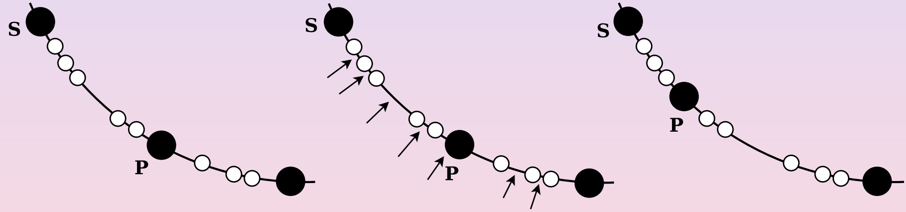
A novel heuristic : *Max Local Throughput* Algorithm



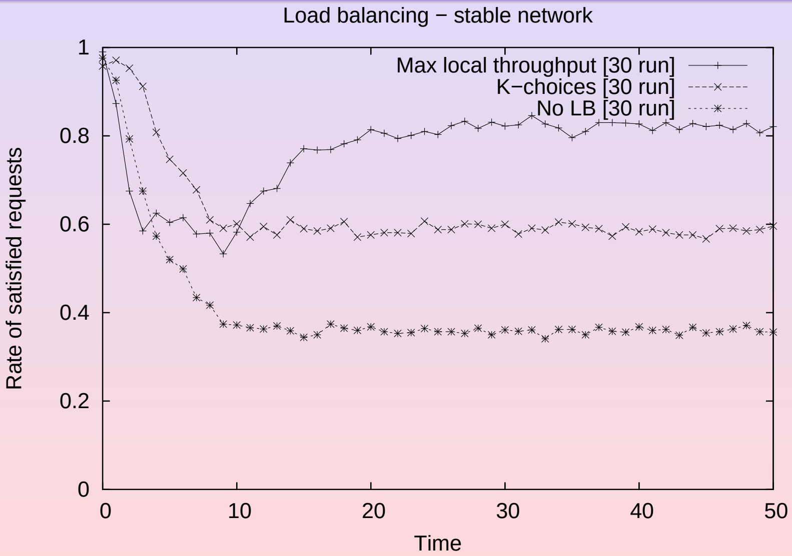
A novel heuristic : *Max Local Throughput* Algorithm



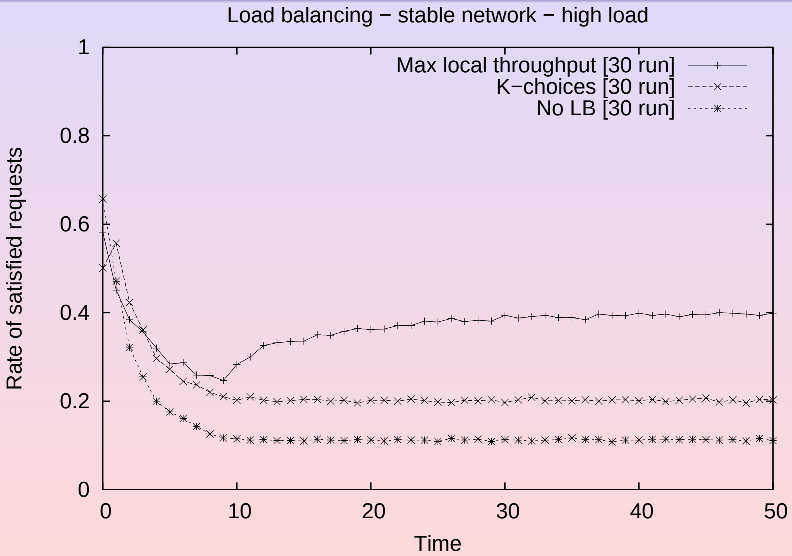
A novel heuristic : *Max Local Throughput* Algorithm



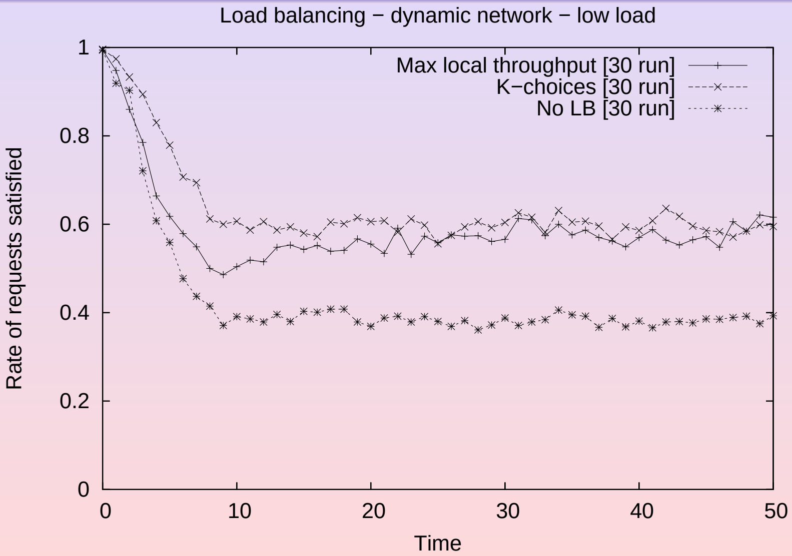
Simulation results



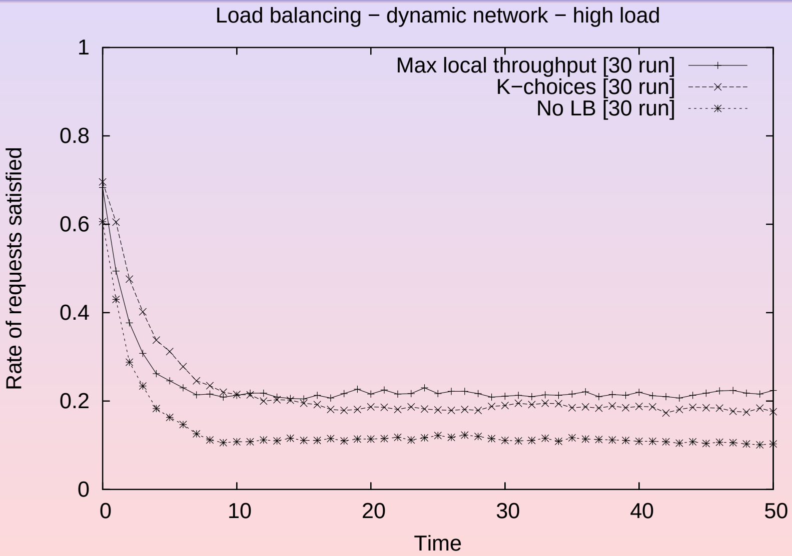
Simulation results



Simulation results



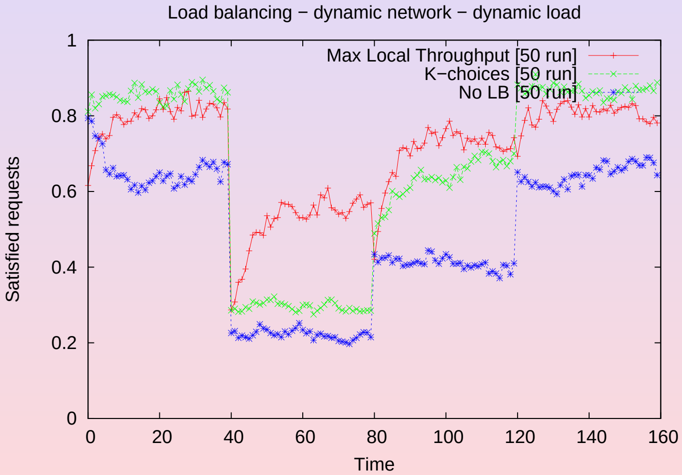
Simulation results



Simulation results

Load	Stable network		Dynamic network	
	Max local th.	K-choices	Max local th.	K-choices
5%	39,62%	38,58%	18.25%	32,47%
10%	103,41%	58,95%	46,16%	51,00%
16%	147,07%	64,97%	65,90%	59,11%
24%	165,25%	59,27%	71,26%	60,01%
40%	206,90%	68,16%	97,71%	67,18%
80%	230,51%	76,99%	90,59%	71,93%

Simulation results



Conclusion and future work

- Contributions
  - A protocol to map a trie on a P2P network
    - Reduction of the architecture cost
    - Better self-containment
  - Load Balancing
    - A novel heuristic
    - Localizing maximizing the throughput
    - Good performance compared to K-choices
  - Dynamic load balancing in tries
- On-going work
  - Prototype development
  - First results are promising