

Optimisation du transport de contenu 5 - Réseaux pair à pair, nommage

Christophe Deleuze
Grenoble INP – ESISAR

Décembre 2017

- **répliquer** les ressources
 - en fct de la demande
 - là où on les demande
- **accéder aux ressources**
 - par leur nom
- **nommer** les ressources
 - indépendamment de leur emplacement
 - indépendamment de leur réplication
 - de façon stable dans le temps

1 / 20

2 / 20

Plan

- ① Le pair-à-pair...
- ② Tables de Hash distribuées
- ③ Le nommage
- ④ Conclusion

Plan

- ① Le pair-à-pair...
- ② Tables de Hash distribuées
- ③ Le nommage
- ④ Conclusion

3 / 20

4 / 20

Napster

- '99 partage de fichiers
- fonctionne en 3 étapes
 - ① indexation : envoi de son index (catalogue) au serveur central
 - ② requête : demande d'un nom de fichier au serveur, retour de la liste des clients possédant le fichier
 - ③ transfert : depuis un des clients

5 / 20

Napster

- transferts pair à pair
 - scalabilité (réplication)
- index sur serveur centralisé
 - regexp possibles
 - pas scalable
 - pas robuste
 - techniquement
 - juridiquement !

6 / 20

Gnutella

- réseau "overlay" – graphe des pairs
- recherche par inondation de requête
 - trop coûteux si inondation complète
 - échec possible si inondation partielle
 - attaques DOS possibles
- téléchargement depuis le pair trouvé

7 / 20

KaZaA/eDonkey

- indexation hiérarchique
 - certains pairs sont supernodes
 - chacun gère un ensemble de pairs "normaux"
 - inondation entre les supernodes
- distribution par **chunks**
 - on télécharge depuis plusieurs pairs

8 / 20

- à chaque contenu est associé un overlay : l'essaïm (*swarm*)
- fichier découpé en blocs de tailles identiques
- *tracker* gère l'essaïm
- fichier `.torrent` contient
 - URL du *tracker*
 - hash crypto de chaque bloc
- client
 - se connecte au *tracker*
 - reçoit liste des clients actifs
 - télécharge blocs depuis les clients
 - fournit les blocs déjà téléchargés à d'autres clients

9 / 20

① Le pair-à-pair...

② Tables de Hash distribuées

③ Le nommage

④ Conclusion

10 / 20

DHT : distributed hash table

- un réseau p2p se caractérise sur deux axes :
 - indexation (comment trouver un document)
 - distribution (comment rapatrier un document)
- pb. p2p : accès (recherche) non scalable
 - serveur centralisé (Napster)
 - inondation (Gnutella)
 - mix (KaZaA)
- solution : DHT
 - $\{(clé, valeur)\}$
 - insertion/recherche/effacement rapide

11 / 20

Content Addressable Network

- espace de coordonnées sur hypercube de dimension d
 - (en fait un hypertore, mais simplifions)
 - clé = point de l'espace (d coordonnées)
 - CAN = ens. de nœuds
 - partition de l'espace entre les nœuds
 - chaque nœud
 - zone (partie de la TH = une portion de l'espace)
 - infos voisins (adr. IP + zone gérée \rightarrow table de routage)
- ⇒ topologie logique (*overlay*)

12 / 20

Content Addressable Network

ex. avec 7 noeuds A,B,...G et $d = 2$

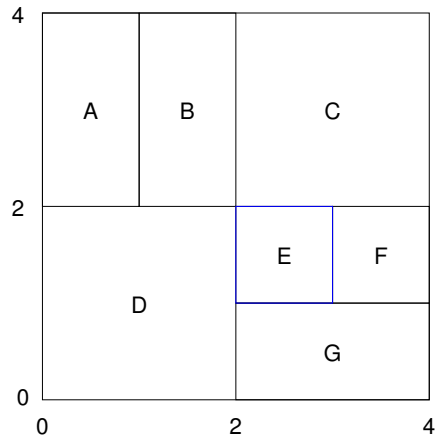


Table de routage de E

E	(2,1) – (3,2)
C	(2,2) – (4,4)
D	(0,0) – (2,2)
F	(3,1) – (4,2)
G	(2,0) – (4,1)

13 / 20

CAN

- Routage
 - n zones, d dim. lg moy ch. = $\frac{1}{4} n^{\frac{1}{d}}$
- Maintenance
 - construction par division de zone
 - mise à jour des voisins uniquement
 - les tables ne mentionnent que les voisins
- Autre approches
 - Chord, Pastry, Tapestry, Viceroy...

15 / 20

Routage dans le CAN

- vers le voisin le plus proche de la clé/point (distance cartésienne) (*greedy forwarding*)
- ex. avec $P = (1.75, 3)$

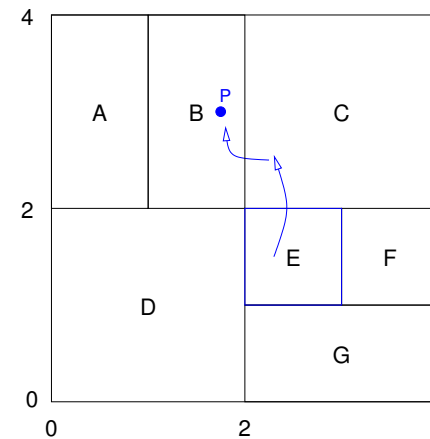


Table de routage de E

E	(2,1) – (3,2)
C	(2,2) – (4,4)
D	(0,0) – (2,2)
F	(3,1) – (4,2)
G	(2,0) – (4,1)

14 / 20

Plan

- 1 Le pair-à-pair...
- 2 Tables de Hash distribuées
- 3 Le nommage
- 4 Conclusion

16 / 20

- adresse
 - identificateur lié à la structure du système
 - purement syntaxique
 - ex. adr. IP
- nom
 - identificateur convivial
 - pour utilisateur
 - contenu sémantique
 - abstractions
- résolution
 - passage d'un nom à une adresse

ex : nom de fichier Unix

17 / 20

- DOI system
 - Digital object identifier
 - *Handle System* : système de résolution
 - `http://dx.doi.org/<le doi>`

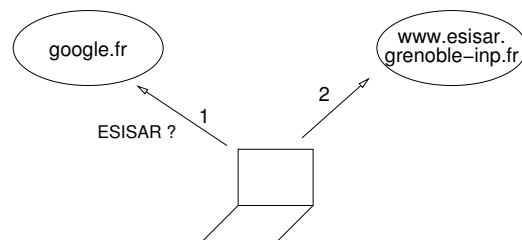
Investigating the Imprecision of IP Block-Based Geolocation
 Book Series Lecture Notes in Computer Science
 Publisher Springer Berlin / Heidelberg
 ISSN 0302-9743 (Print) 1611-3349 (Online)
 Volume Volume 4427/2007
 Book Passive and Active Network Measurement
DOI 10.1007/978-3-540-71617-4
 ISBN 978-3-540-71616-7
DOI 10.1007/978-3-540-71617-4_26

- services de redirection
 - go.to
 - tinyurl ...

18 / 20

Moteurs de recherche

- beaucoup de gens ne tapent plus les URL
- ne posent plus de marque-page...
- ... mais passent par google!
- "I feel lucky"
 - = service de résolution
 - + pages en cache chez Google!



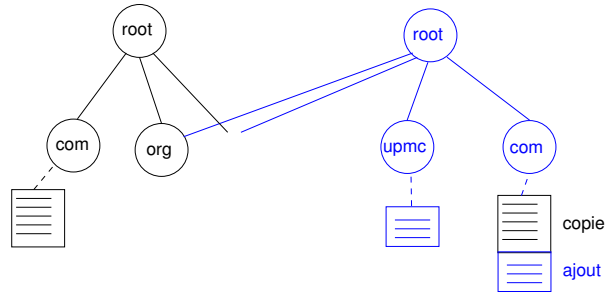
19 / 20

Nommage : aspects sociaux

- les noms ont un aspect sémantique...
- source de grande valeur
 - marques déposées
 - conflits DNS
- la gestion du DNS est contestée
 - racines alternatives
 - rfc2826 "IAB Technical Comment on the Unique DNS Root" (mai 2000)

20 / 20

- pointent sur la hiérarchie officielle...
- ...plus d'autres noms de domaines
- ex. nouveau TLD `.upmc` et version *étendue* de `.com`
- résolveur doit utiliser la racine bleue `root`



21 / 20

Plan

- ① Le pair-à-pair...
- ② Tables de Hash distribuées
- ③ Le nommage
- ④ Conclusion

23 / 20

- je résouds `wwx.google.fr`
 - le résolveur devrait répondre NXDomain (no such domain)
 - il me répond l'adresse IP d'un serveur web qui me propose un lien vers `http://www.google.fr` (et affiche une pub)

 - je résouds `www.mechantspirates.com`
 - le résolveur ne répond pas la vraie adresse...
 - mais celle d'un serveur web qui me met en garde
- pb : si le *user-agent* n'est pas un navigateur

22 / 20

Réseaux de contenus

- Une infrastructure fournissant
 - réplication
 - redirection/indexation/nommage
 - distribution
- possédée par (au choix) :
 - les opérateurs de réseau (proxies)
 - les fournisseurs de contenu (fermes, CDN)
 - les utilisateurs! (p2p)
- routage au niveau de la couche application
 - ➡ Content Centric Networking

24 / 20