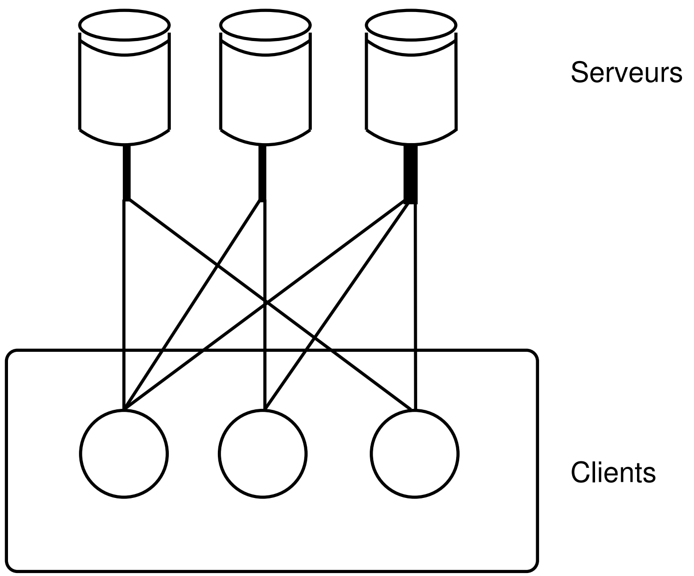
Caches web

Olivier Aubert

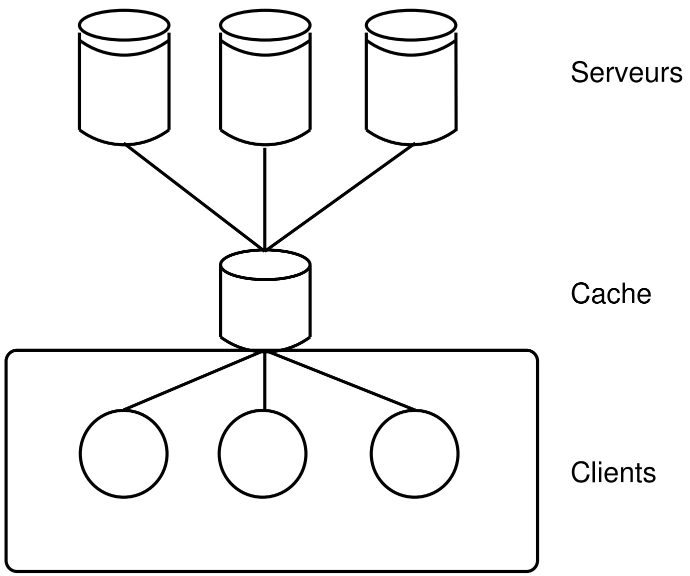
Liens

- http://mqdoc.lasat.com/online/courses/caching/ (prise en compte des caches dans la conception de sites)
- http://mqdoc.lasat.com/online/courses/proxyserver/
- http://www.web-caching.com/mnot_tutorial/

Problématique



Problématique



Intérêts pour l'utilisateur

- Réduction des temps d'accès :
  - Documents cachés et bande passante économisée
  - Distinction temps de latence et temps de transfert
- Services supplémentaires :
  - Filtrages divers (antivirus, anonymisation)

Intérêts pour le fournisseur d'accès

- Diminution du trafic
- Réduction de la congestion du réseau
- Amélioration des performances
- Économies sur les équipements réseau
- Possibilité de filtrage (antivirus)
- Possibilité de restreindre les accès (enfants, classes d'utilisateurs, ...)

Intérêts pour le fournisseur de services

- Prise en charge d'une partie du trafic par les caches ⇒ réduction de la charge des serveurs
- Utilisation de caches inverses pour alléger la charge des serveurs
- Augmentation de la robustesse du service (en cas de coupure du serveur)

Services supplémentaires

- Anonymisation : `anonymize_headers` dans Squid
- Suppression des bannières publicitaires : `http://internet.junkbuster.com/`
- Préchargement : *CacheFlow*
- Fonctionnement déconnecté : *WWWOffle*
- Sécurité : `http://www.antivirus.com/`

Inconvénients

- Documents pas toujours à jour
- Fonctionnement pas toujours transparent (ftp, ...)
- Performances pas forcément améliorées (sous-dimensionnement)
- Point de panne unique
- Intérêt diminué par l'augmentation du nombre de sites dynamiques ou personnalisés (cookies)
- Augmentation de la difficulté du comptage d'accès
- Problèmes juridiques (copyright)

Types de caches

- Caches personnels (navigateurs)
- Caches institutionnels
- Caches régionaux/nationaux

Fonctionnalités des caches web

- filtrage des documents (contrôle d'accès, cachabilité)
- préchargement des documents
- coopération entre caches
- gestion de la cohérence
- remplacement des documents
- accès au ressources (stockage)

Filtrage

- Sélection des utilisateurs pouvant utiliser le service de cache
- Possibilité de définir des classes d'utilisateurs, des horaires d'utilisation, des limitations de bande passante
- Vérification de la cachabilité d'un document

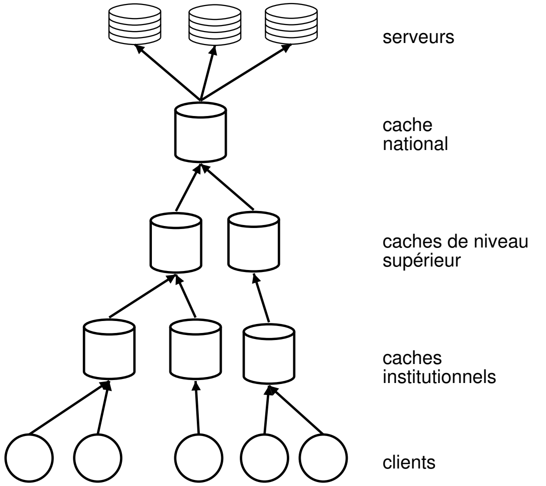
Préchargement de données

- Anticipation des requêtes futures
- Plusieurs modes :
  - effectué par les clients pour les documents provenant des serveurs (avant la diffusion des caches)
  - effectué par les caches pour les documents provenant des serveurs (plus intéressant, prévisions plus fiables)
  - effectué par les clients pour les documents provenant des caches (dans le cas de connexions finales à faible débit (modems))

Architecture

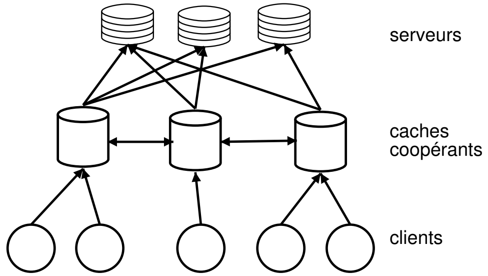
- Architecture hiérarchique
- Architecture distribuée
- Architecture hybride

Architecture hiérarchique



- Utilisation plus réduite de la bande passante
- Taux de réussite cumulés

Architecture distribuée



- Utilisation plus équilibrée des liaisons réseau

Coopération

- Localisation de l'information *a posteriori* : ICP, HTCP, CRISP
- Localisation de l'information *a priori* : CARP, Digests

ICP

- Internet Cache Protocol, RFC2186
- Un des premiers protocoles de coopération
- Basé sur UDP (estimation des performances du réseau)
- Problèmes
  - faux hits (informations incomplètes [entêtes])
  - Redimensionnement
  - Sécurité
  - Introduction de délais

HTCP

- `HyperText Caching Protocol`
- Dérivé de ICP
- Ajout notamment des entêtes des documents
- Possibilité de contrôle du fonctionnement de caches distants

CRISP

- Ensemble de caches autonomes
- Répertoires partagés via un service de localisation commun
- Possibilité de répliquer les données

CARP

- *Cache Array Routing Protocol*
- Répartition des URL par l'utilisation d'une fonction de hashage
- Pas vraiment de coopération entre caches
- Augmentation des taux de hits
- Mise en œuvre sur un proxy d'entrée ou sur chaque client

Digests

- Envoi périodiques de résumés condensés des contenus des caches
- Économie de bande passante (50% wrt ICP), réduction de la charge CPU
- Possibilité de ne transmettre que les deltas
- Problèmes
  - taille des résumés
  - faux hits ou faux misses

Politiques de cohérence

- Objectif : s'assurer de la validité des données cachées par rapport aux données originales
- Modèles de cohérence : forte, faible
- Plusieurs techniques de maintien

Maintien de la cohérence

- Vérification périodique par le cache
- Notification d'invalidation par les serveurs (directe ou en piggybacking)
- Estimation (par le cache ou par le serveur) du TTL (Time-To-Live)
- Utilisation des requêtes `If-Modified-Since`

Politiques de remplacement

- Détermine les documents à supprimer pour faire de la place
- Utilisation de divers critères :
  - Traditionnels : LRU, LFU
  - Politiques à clés : Size, Lowest Latency First, LRU-MIN
  - Politiques à fonctions de coût : GreedyDualSize, Hybrid, Lowest Relative Value
- Difficultés d'optimiser plusieurs critères différents
- Importance du facteur taille

Accès aux ressources

- Placement coopératif
- Prise en compte de la localité de référence
- Partitionnement de l'espace de stockage

Mesure des performances

- Définition des critères
- Outils de mesure

Critères généraux

- Vitesse
- Capacité de passage à l'échelle (*scalability*)
- Robustesse
- Utilisateur : temps de latence, temps de chargement, fraîcheur des documents
- Fournisseur : bande passante économisée

Taux de réussite

- *Hit Ratio*
- Nombre de hits sur nombre de requêtes
- Limite supérieure (cache de taille infinie) : 30 à 50%
- Pondération par la taille des documents (*Byte Hit Ratio*)

Critères temporels

- *Delay Savings Ratio* : amélioration de la vitesse de chargement
- Nécessité de prise en compte des temps de réaction

Métriques d'exploitation

- *Scalability*
- Comportement lors des surcharges
- Mesure des économies réalisées (bande passante, équipements, financières)
- Coût d'administration

Outils de mesure

- Simulation
- Mesures par génération de trafic artificiel
- Analyse de fichiers de données

Outil : simulation

- Saperlipopette!
- Simulateur de caches web distribuées développé à l'INRIA
- Objectif : déterminer les architectures de caches les plus adaptées
- Fonctionnement : simulation à événements discrets. Utilisation de traces capturées.

Outil : analyse

- *squeezer* : analyse des fichiers de log de Squid, avec un guide d'interprétation.
- *calamaris* : analyse des fichiers de log de plusieurs caches. Statistiques synthétiques du trafic observé.

Outil : trafic artificiel

- *Polygraph!*, outil répandu de mesure des performances des caches web.
- Génération de requêtes suivant un profil défini (distribution des objets, localité temporelle, cachabilité, ...)
- Permet d'effectuer des évaluations de scénarios futurs (accroissement du nombre d'utilisateurs ou des types de données, ...)
- Utilisé pour effectuer régulièrement un benchmark multi-constructeurs.

Exemples de caches

- Squid, Apache
- CacheFlow (matériel, préfetching)
- CISCO Cache Engine (intégration aux routeurs CISCO)
- CacheQube (administration simplifiée)
- Dynacache (InfoLibria)
- Internet Security and Acceleration Server (ISA) : Microsoft
- NetCache (NetAppliance)
- Netscape Proxy Server
- Oracle9i Application Server Cache (statique et dynamique)
- WWWOffle : fonctionnement déconnecté