



Désignation dans les systèmes répartis

***Michel RIVEILL
Bull-IMAG/Systèmes
2, rue de Vignate
38610 Gières***

Michel.Riveill@imag.fr

Juillet 1993



Exemple introductif

❖ Problème :

- Depuis un processus exécuté sur un site,**
- Accéder à un fichier situé sur un site distant**

❖ Principe de trois solutions :

- Transfert sur le site local**
- Accès distant, localisation explicite**
- Accès distant, localisation transparente**

❖ Mise en évidence des problèmes de la répartition sur ces trois solutions



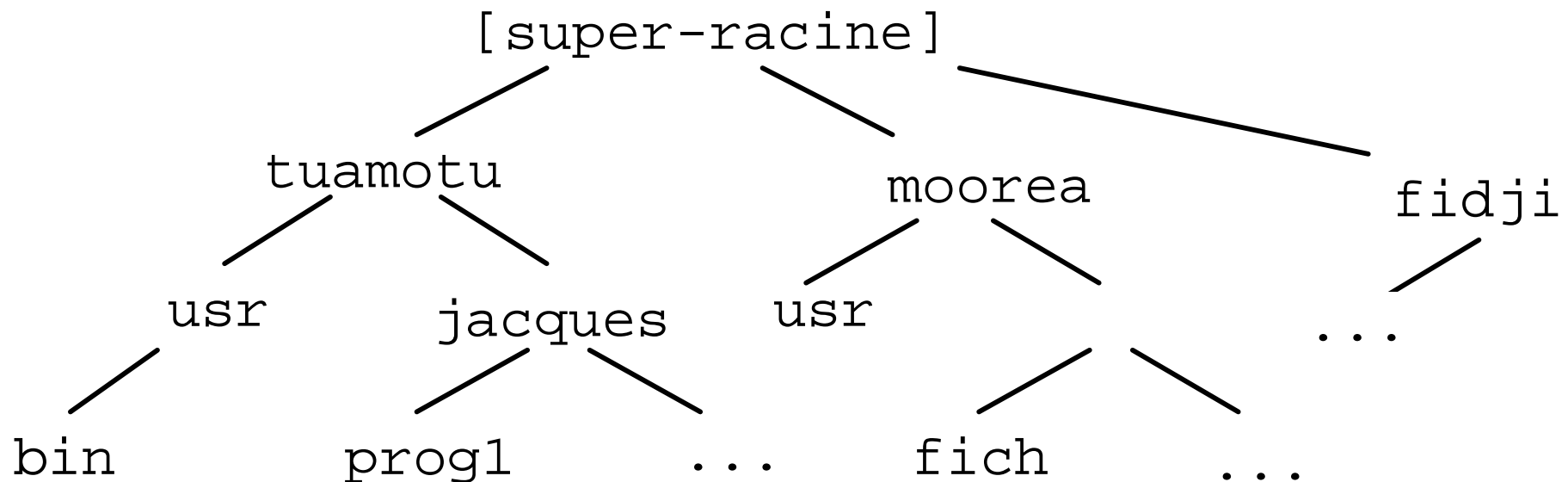
Commentaires

- ❖ **Solution très partielle au problème posé**
- ❖ **Existence de deux exemplaires distincts**
- ❖ **Cohérence à la charge de l'utilisateur**
- ❖ **Problème du partage entre clients multiples**
- ❖ **Duplication possible des protocoles (local vs “bout-en-bout”)**

Solution 2 : Système de gestion de fichiers

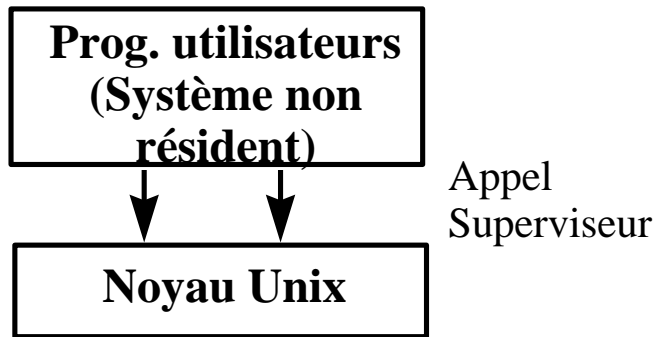
Accès à distance - localisation explicite (Newcastle Connection)

- ❖ **Principe**: réunion des espaces de noms par une “super-racine”

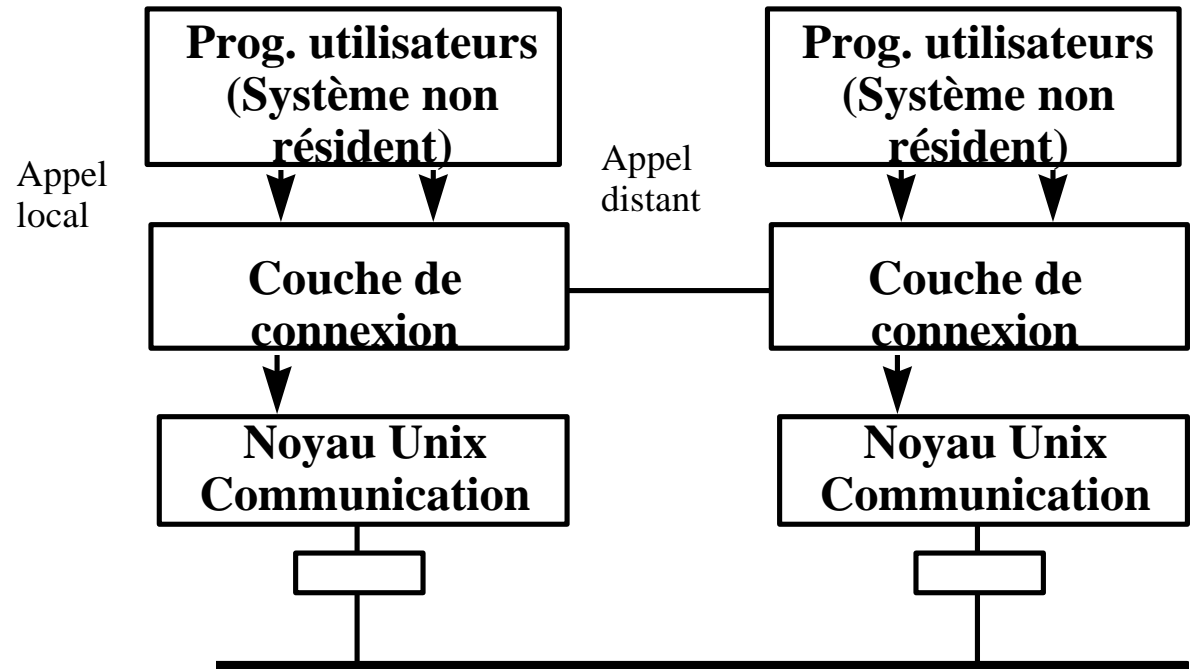


Réalisation

a) Système Unix



b) Newcastle connexion





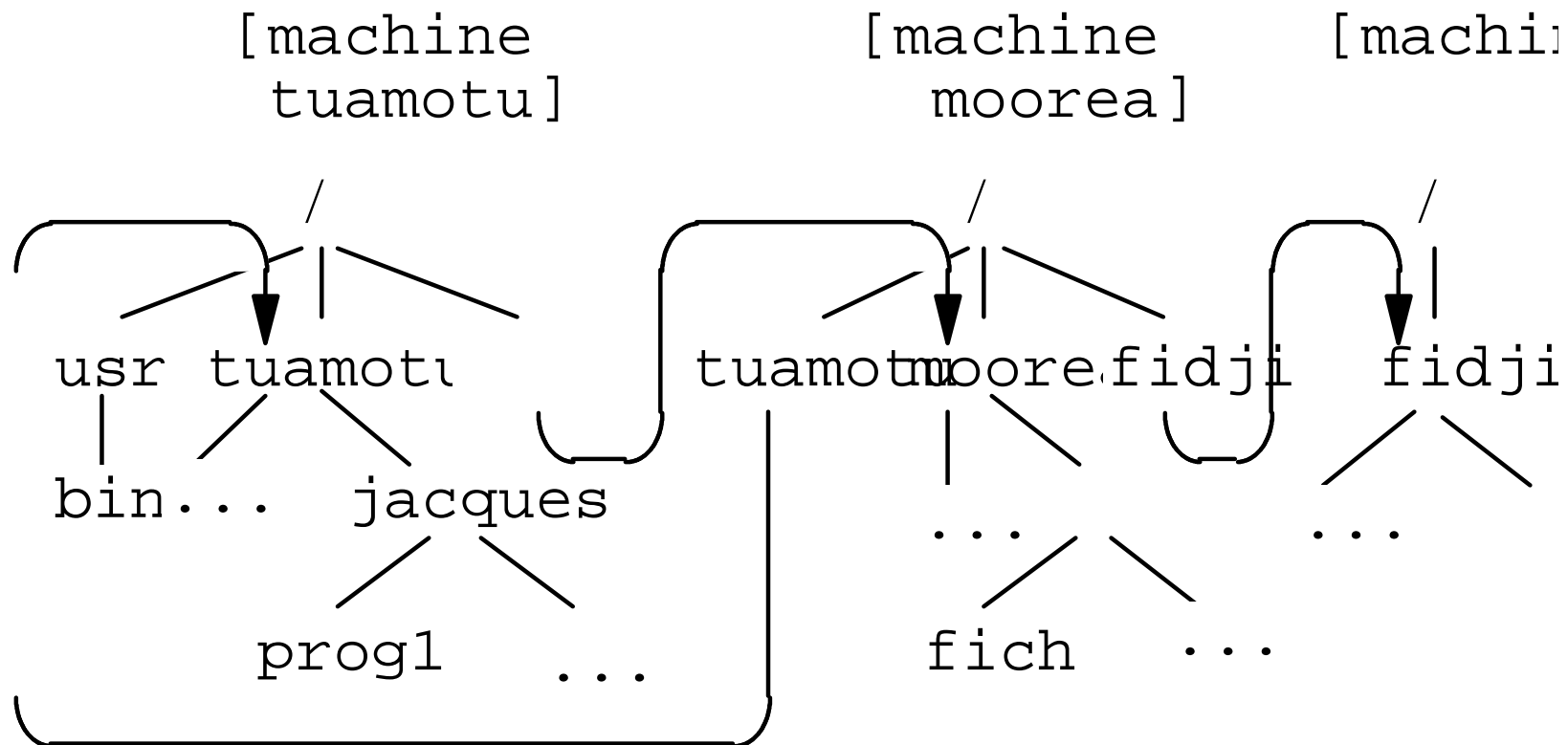
Commentaires

- ❖ **Réalisation assez aisée, mais transparence limitée**
- ❖ **Problèmes pour le partage (mais pas plus que Unix)**
- ❖ **Problèmes d'administration (gestion des droits d'accès)**

Solution 3 : système de gestion de fichiers

Accès à distance - localisation cachée (NFS)

❖ **Principe: “Montage” logique d’une sous-arborescence distante**





Réalisation et commentaires

❖ Réalisation

- Exportation explicite nécessaire (pour protection)**
- Un niveau supplémentaire d'indirection (solution classique)**
- Transfert d'information sur le principe client-serveur**
- Appel de procédure à distance**
- Accélération des accès par caches (client et serveur)**

❖ Commentaires

- Accès "transparent"**
- Séparation des fonctions du client et de l'administrateur**
- Pas de masquage des défaillances du serveur**
- Problèmes de partage**
 - logique (comme précédemment)**
 - cohérence des caches multiples**



Mise en œuvre d'une exécution répartie

❖ Désignation des entités

- Noms, accès, liaison**
- Structure et évolution de l'espace des noms**
- Service de désignation**

❖ Structures élémentaires d'exécution

- Modèle client-serveur**
- Autres schémas d'exécution**
 - ◆ migration, diffusion**
 - ◆ partage d'objets**

Rappel des notions de base

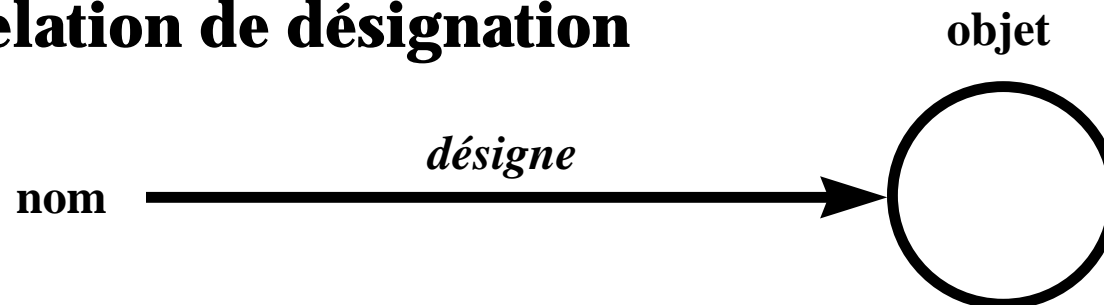
❖ **Définition d'un nom**

- **information associée à une entité**
- **pour l'identifier (la distinguer des autres)**
- **pour l'atteindre et l'utiliser (voie d'accès)**

❖ **Niveaux de désignation**

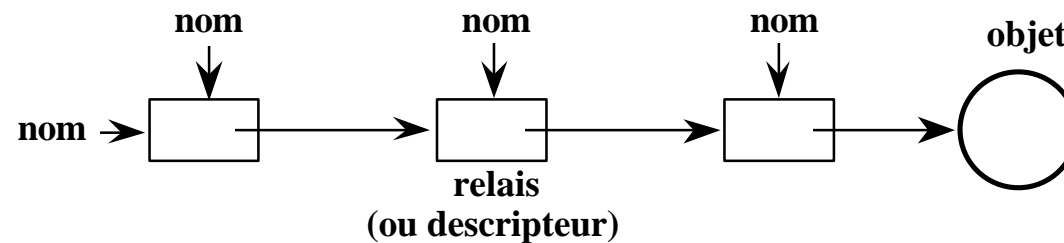
- **pour les utilisateurs : noms externes - identificateurs**
- **pour le système : noms internes - descripteurs**

❖ **Relation de désignation**



Relation de désignation

❖ Chaîne d'accès



❖ Liaison : établissement de la chaîne d'accès

- Statique ou dynamique
- Substitution ou chaînage

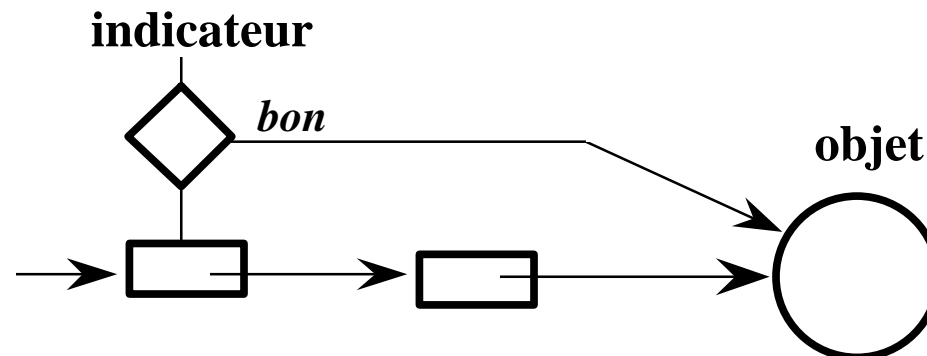
Relation de désignation

❖ Contextes et environnements

- Restriction de l'espace des noms
- Protection
- Efficacité (accélère la recherche)

❖ Mécanismes d'accélération

- Indicateur (accès rapide si correct, vérification automatique)
- Cache (d'objets ou d'indicateurs)





Mode de désignation

❖ **Noms externes (symbolique)**

- interprétables par utilisateurs, applications (désignation d'une grande variété d'objets)
 - ♦ noms de fichiers `imag:/users/guide`
 - ♦ noms de sites sur un réseau `bounty.imag.fr`

❖ **Noms internes (chaîne de bits)**

- interprétables par le système (unicité - efficacité)
 - ♦ adresses physiques en mémoire
 - ♦ adresses de sites sur un réseau `192.44.69.13`
 - ♦ noms universels `<numéro de site><estampille locale>[nb aléatoire]`

❖ **Descripteurs et relais**

- descripteurs de fichiers (i-nodes Unix, v-nodes NFS)
- portes : désignation indirecte pour un processus
 - association dynamique processus-porte
 - remplacement, reprise
 - mobilité géographique des processus et portes



Désignation et localisation

- ❖ **Distinction entre noms "locaux" et "globaux"**
 - dépendent ou non d'un contexte d'évaluation
- ❖ **Distinction entre noms "impurs" et "purs"**
 - contiennent ou non une information sur la localisation

- ❖ **S'applique aux noms externes ou internes**
- ❖ **S'applique aux liens entre contextes**
 - purs : liens symboliques (voir SGF)
 environnement de résolution

 - impurs : liens de poursuite
 environnement de recherche



Noms "purs ou impurs"

❖ Noms "purs"

- immuables, uniques**
- interprétation**
 - diffusion coûteuse**
 - techniques d'accélération**
 - contexte restreint**
 - indications (validité non garantie)**
 - mise en cache des références récentes`**

❖ Noms "impurs"

- en général valables dans un contexte**
- problème de l'évolution**
 - liens de poursuite**
 - gérant de localisation**
 - techniques mixtes**



Service de désignation

❖ Fonctions

- **Réaliser la correspondance entre noms et objets (souvent via des noms internes)**
 - ♦ **liaison nom externe-nom interne**
 - ♦ **liaison nom interne-localisation**
 - ♦ **Ces deux fonctions peuvent être :**
 - ♦ **dans même service (Unix, DEC)**
 - ♦ **deux services distincts (V-kernel, Sprite)**
- **Répertoire d'informations sur les services**
 - ♦ **possibilité de recherche associative**

❖ Réalisation

- **Serveur de désignation (ou de noms)**
- **Serveurs coopérants**
- **Nom du serveur connu a priori ou obtenu facilement (à l'initialisation ou la connexion)**



Problèmes spécifique de la répartition

❖ L'espace des noms a une grande taille

- **Sur un réseau à grande distance**
 - ♦ **désigner toutes les machines, clients (boites aux lettres ...)**
- **Sur un réseau local (granularité plus fine)**
 - ♦ **désigner tous les processus, les fichiers, ...**
- **Organisation et personnalisation**

❖ L'espace des noms évolue dynamiquement

- **Création de nouvelles entités**
- **Adjonction de nouveaux composants (machines, sous-réseaux, ...)**
- **Réorganisation de la structure interne (contraintes d'administration, ...)**
- **Extension et restructuration de l'espace des noms**



Les problèmes de la répartition (suite)

- ❖ **La “transparence” est recherchée**
 - **Indépendance nom-localisation (possibilité de migration)**
 - **Conséquences:**
 - ◆ **liaison dynamique fréquente**
 - ◆ **réalisation par un service identifié**
- ❖ **La disponibilité du service est critique**
 - **Redondance pour résister aux défaillances partielles**
- ❖ **Pas spécifiques mais important**
 - **Administration (réparties ?)**
 - **Efficacité : service très sollicité**

Désignation réparties

Structure et évolution de l'espace des noms

❖ Organisation hiérarchique

– Structure arborescente de l'espace des noms

– Exemples:

◆ **Système de désignation domainisé pour réseau**

– arisia.xerox.com

– vlsivax.cs.cmu.edu

– rangiroa.imag.fr

◆ **Montage logique (NFS : mount/export)**

◆ **Racine virtuelle (Newcastle connexion)**

◆ **Tables de préfixes (Sprite, V-System)**

❖ Autre approche : noms "plats"

– **Fonction de hachage :** F(nom) -> Numéro du serveur

– **Expressions régulières :** l?y* -> Numéro du serveur



Désignation réparties

Structure et évolution de l'espace des noms

❖ Modes d'évolution

- Réunion d'espaces indépendants
 - ◆ Création d'une super-racine virtuelle
 - ◆ Montage logique
- Ajout d'éléments
 - ◆ Extension sans changement de racine
 - ◆ Extension “par le haut” (modifie les noms absolus)
- Modifications de structure

❖ Mécanismes d'accélération

- Maintien du contexte courant
- Création de liens, indicateurs, caches



Désignation réparties

Techniques de base pour les serveurs de désignation

❖ Administration hiérarchique de l'espace des noms

- Découpage en zones gérées par des serveurs distincts
(Ce découpage coïncide ou nom avec celui des noms)**
- Répartition de la charge**
- Répartition de l'autorité administrative**

❖ Duplication partielle

- Réduction des temps d'accès**
- Disponibilité en cas de panne**

❖ Cohérence faible des copies multiples

❖ Utilisation de caches et d'indicateurs

- Amélioration des performances pour les cas usuels**



Exemple d'un service de désignation

Service global de DEC [Lampson 86]

❖ Objectifs

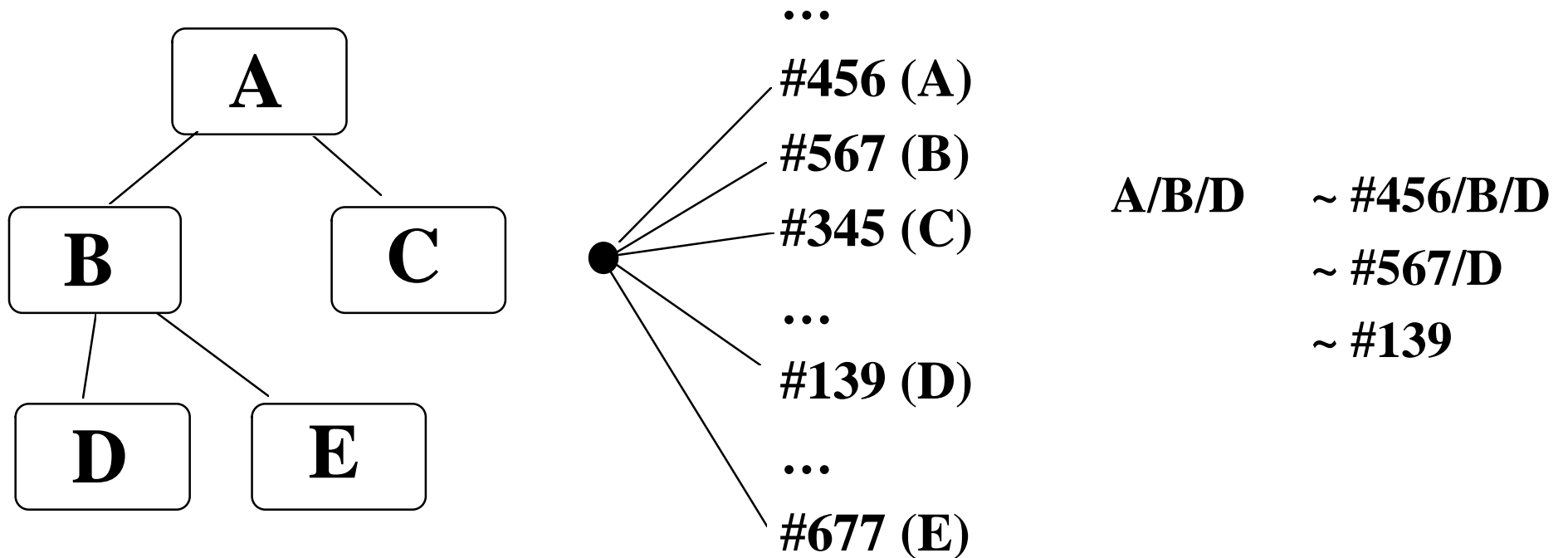
Réaliser un service de désignation

- ♦ **de très grande taille**
- ♦ **de très grande durée de vie**
- ♦ **permettant des restructurations importantes de l'espace des noms**

❖ Espace des noms

- **Vu externe : nom hiérarchique (noms de catalogue + noms locaux)**
- **Vu interne : désignation universelle (DirId) pour tout catalogue**

Service de désignation de DEC (suite)



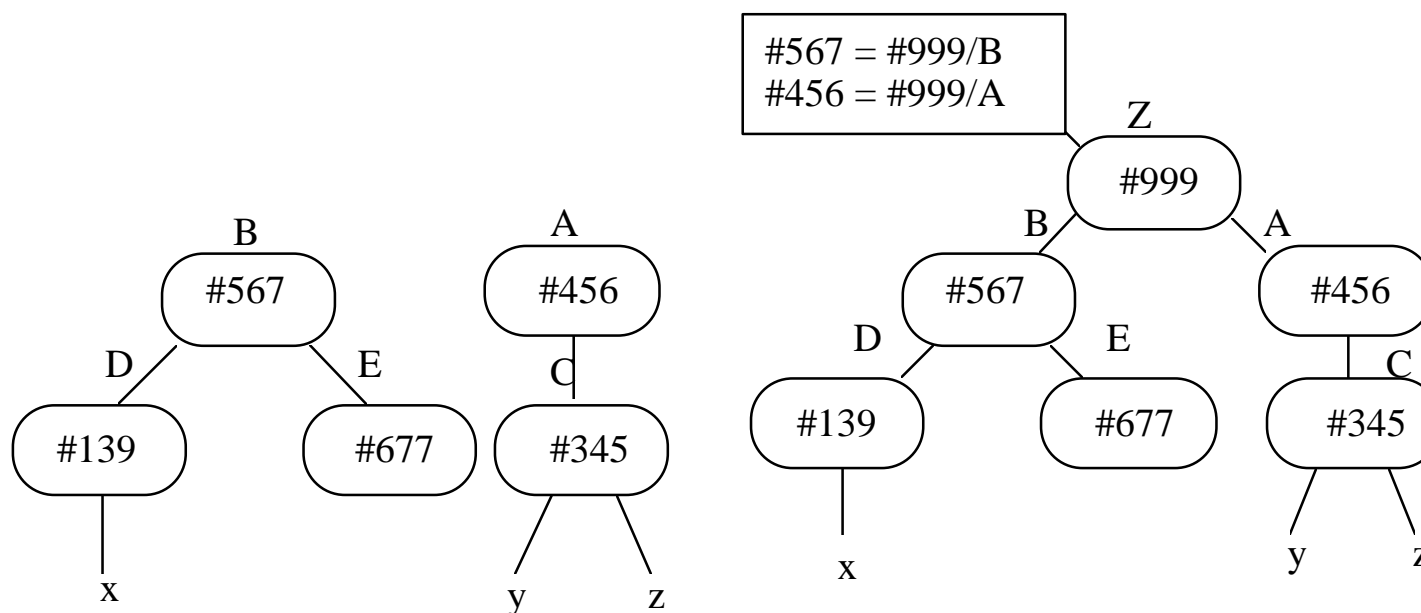
- **Avantage : restructuration / combinaison simplifiées**
- **Problème : retrouver un catalogue à partir du DirId (recherche dans un très grand espace plat)**

Service de désignation de DEC (suite)

Réorganisation de l'espace des noms

On suppose résolue la recherche des catalogues

❖ Regroupement

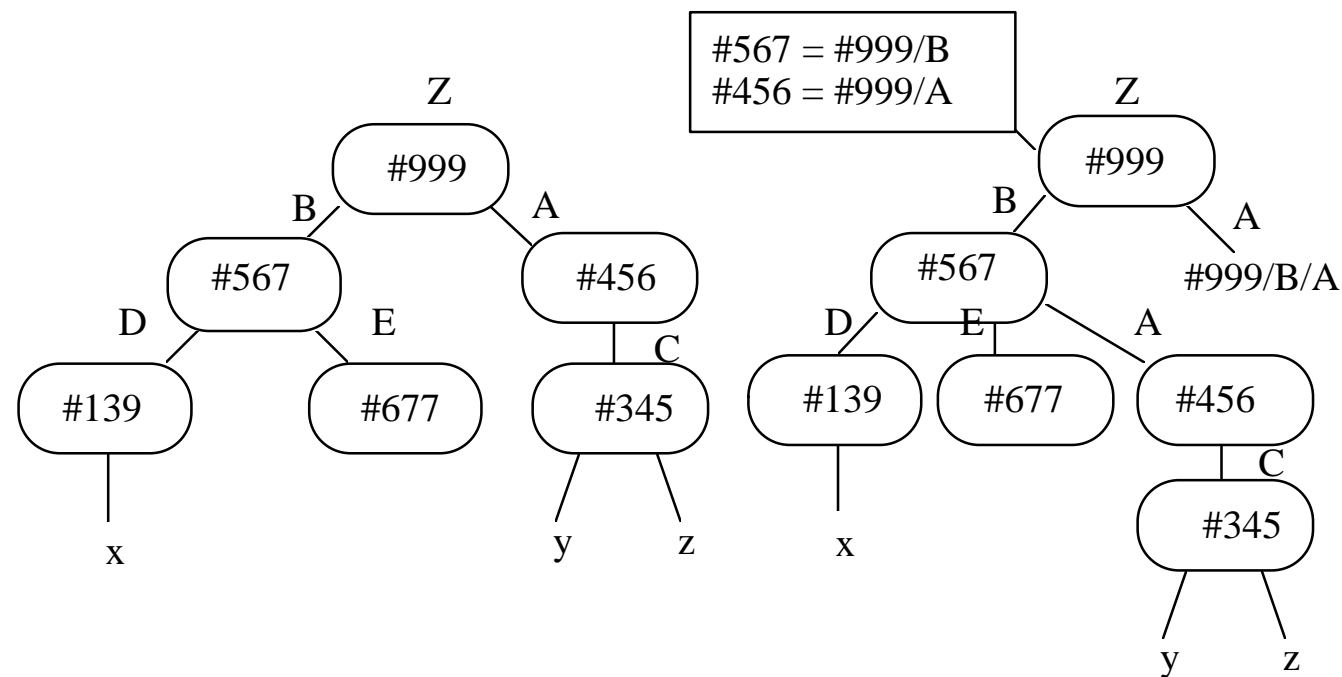


Service de désignation de DEC (suite)

Réorganisation de l'espace des noms

❖ **Déplacement d'un sous-arbre**

Techniques utilisées : accourci d'adressage + cache





Service de désignation de DEC (suite)

Localisation des catalogues

❖ **Fonction : trouver un catalogue à partir de son DirId**

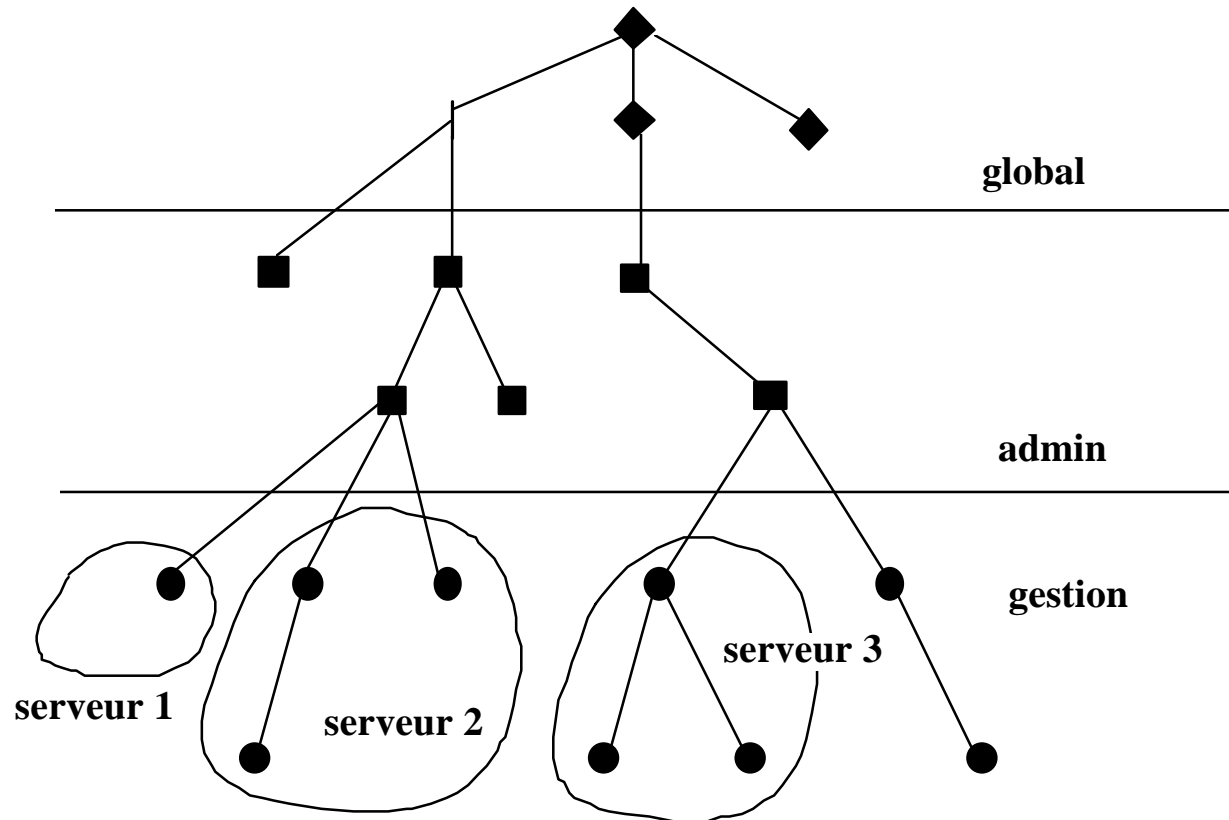
❖ **Principe :**

- **Les catalogues sont stockés dans des serveurs**
- **Les serveurs sont eux-mêmes identifiés par le service de noms (invariants de structure à maintenir pour éviter les boucles!)**
- **Chaque catalogue est stocké dans plusieurs serveurs avec cohérence faible entre les copies (resynchronisation périodique)**
- **Mécanismes d'accélération**
 - ◆ **Liens inverses**
 - ◆ **Caches pour localisation des catalogues**
 - ◆ **Racine accessible (directement ou non depuis tout serveur)**

Exemple d'un service de désignation

V-kernel, Stanford [Cheriton 89]

- ❖ **Idée : découpler structure de l'espace des noms et processus de localisation**





Service de désignation du V-kernel

❖ **Problème : identifier le serveur qui gère un objet donné**

❖ **Solution : “cache des préfixes”**

préfixe d'un nom => IdServeur

on cherche le plus long préfixe présent dans le cache

si succès (serveur trouvé) : recherche avec reste du nom

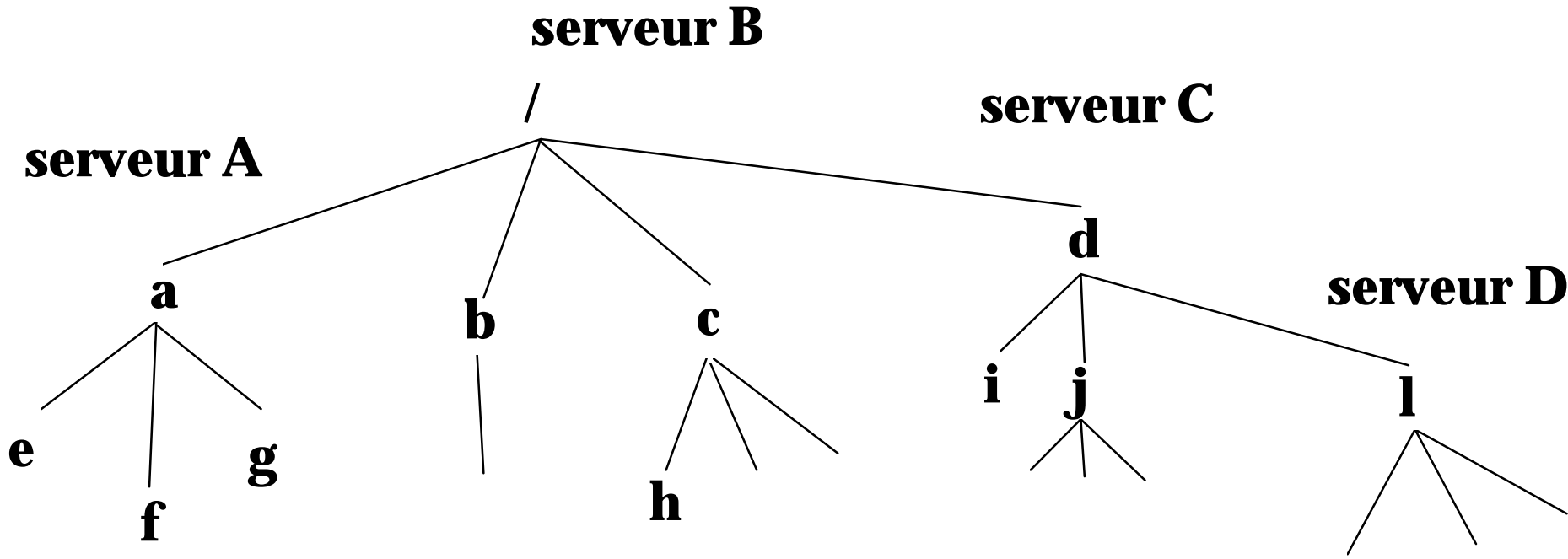
(ex : a/b/c/d/e - préfixe = a/b/c ; recherche sur d/e)

si succès partiel => groupe de serveurs

si échec => recherche par diffusion, distances croissantes



Table de préfixes



<i>Table de préfixes</i>	
/	B
/a	A
/d	C
/d/k	



Le service de désignation de Guide

❖ Objectifs

- Indépendant du système**
- Extensible**
- Efficace**
- Tolérant aux défaillance**

❖ Hypothèses

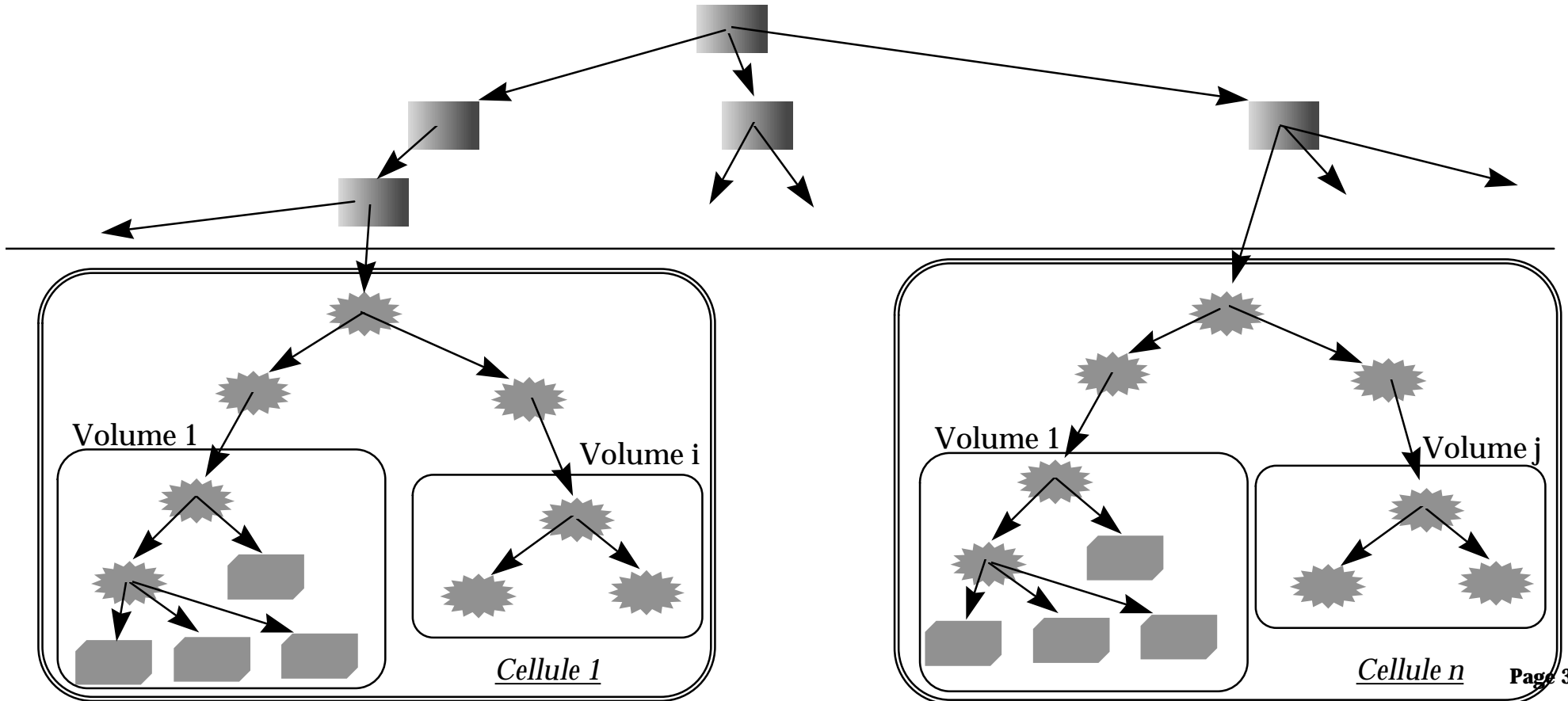
- Construit comme un service du système**

❖ Espace des noms :

- construction d'un arbre de noms unique à base d'objets répertoires de deux catégories :**
 - ♦ Répertoires de niveau global dont les entrées désignent :**
 - d'autres répertoires globaux ou des répertoires racines de cellules**
 - ♦ Répertoires de niveau local dont les entrées désignent :**
 - d'autres répertoires locaux ou des objets**

Architecture

Répertoires globaux (inter-cellules)
 Répertoires locaux (intra-cellules)
 Objets



Extensibilité

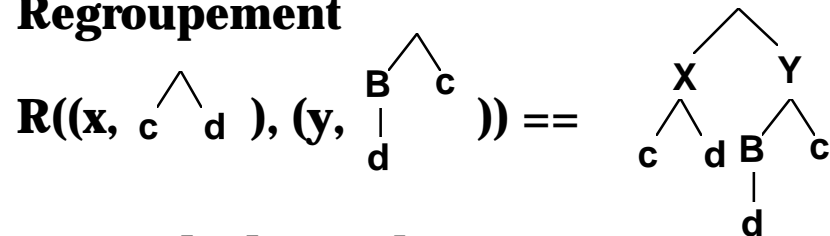
❖ Réalisation de l'extension par le haut

- Opérateurs de regroupement et de restructuration des arbres au niveau global
- Mécanismes de préservation de la validité des anciens noms (Lampson)

❖ Personnalisation des noms

- Vue "personnelle" de la désignation primitive construite par composition à l'aide des opérateurs :

◆ Regroupement



◆ Union de deux arbres

◆ Filtre à l'aide d'une expression régulière



Efficacité

❖ Maintenir des caches de noms

- Au niveau utilisateur : indicateur**
- Au niveau de la cellule : date d'expiration**

❖ Cache utilisateur

- Exploiter la propriété de localité des nos exhibée par l'utilisateur**
- Cohérence**
 - ◆ Pas de cohérence (indicateurs)**

❖ Cache de la cellule

- Limiter le coût des opérations de recherches inter-cellules**
- Exploiter la localité des noms au niveau d'une cellule**
- Cohérence**
 - ◆ Par dates d'échéances**



Tolérance aux défaillances

❖ Deux objectifs

- Autonomie

- ♦ Des volumes : algorithme de duplication**
- ♦ Des cellules : racine unique**

Trois modes de fonctionnement

- mono-site / cellule / intercellule**

→ Permet le fonctionnement isolé d'une partie du système

- Duplication et stratégie de placement des répertoires

deux niveaux

- Inter-cellule : fiabilité faible, accès lent**
- Intra-cellule : bonne fiabilité, accès rapide**

→ Duplication et mises à jour appropriées à chaque niveau

- Cohérence faible entre les copies des répertoires de niveau global**
 - Gestion d'un anneau de copies et propagation périodiques des modifications**
- Cohérence forte au niveau local**
 - Une copie maîtresse**



Bibliographie

- ❖ **B.W. Lampson,**
"Designing a Global Name Services", pp. 1-10,
Proc. of the 5th Symp. on Princ. of Distr. Comp., Calgary, Canada, August 1986
 - ***Désignation dans les systèmes de très grande taille***
 - ♦ ***Extension par le haut avec préservation des anciens noms***
 - ♦ ***Mise à jour avec contraintes relâchées***

- ❖ **D.E. Comer and L.L. Peterson,**
"Understanding Naming in Distributed Systems"
Distributed Computing, pp. 51-60, 1989
 - ***Modélisation du problème de la désignation***

- ❖ **B. Welch and J.K. Outerhout**
"Prefix Tables: a Simple Mechanism for Locating Files in a Distributed Systems"
Proc. of the 6th I.C.D.C.S, pp. 523-530, Cambridge, Massachussetts, mai 1986
 - ***Résolution de noms dans un système possédant***



Bibliographie (suite)

- ❖ **D.R. Cheriton and T.P. Mann**
"Decentralizing a Global Naming Service for Improved Performance and Fault Tolerance"
ACM Trans. on Comp. Systems, vol 7(2), pp. 147-183, mai 1989
 - ***Service de désignation de très grande taille basé sur une conception originale***
 - ***Amélioration des performances et tolérance aux pannes***

- ❖ **B. Clifford Newman**
"Scale in Distributed Systems"
Readings in Distributed Computing Systems, IEEE Computer Society Press, 1992
 - ***Etude du facteur d'échelle sur le service de désignation***

- ❖ **K.W. Shirrif and J.K. Ousterhout**
"A Trace-Driven Analysis of Name and Attribute Caching in Distributed System"
Proc. of the 6th I.C.D.C.S, Cambridge, Massachusetts, mai 1986
 - ***Quelques mesures permettant de configurer les caches du service de désignation***

- ❖ **P.K.. Shina, M. Maekawa and K. Shimizu**
"Improving the Reliability of Name Resolution Mechanism in Distributed Operating Systems"
IEEE Proc. on Comp. Distr. Syst., pp. 589-596, 1992
 - ***Etude des mécanismes de tolérance aux pannes***